

w81

**METHOD AND DEVICE FOR EXTRACTING FEATURE CHARACTER STRING, METHOD AND DEVICE FOR RETRIEVING PSEUDO DOCUMENT USING THEM, STORAGE MEDIUM FOR STORING FEATURE CHARACTER STRING EXTRACTING PROGRAM AND STORAGE MEDIUM FOR STORING PSEUDO DOCUMENT RETRIEVING PROGRAM**

Patent Number: JP11338883  
Publication date: 1999-12-10  
Inventor(s): MATSUBAYASHI TADATAKA; TADA KATSUMI; OKAMOTO TAKUYA; SUGAYA NATSUKO; KAWASHITA YASUSHI  
Applicant(s):: HITACHI LTD  
Requested Patent: ☐ JP11338883  
Application Number: JP19980148721 19980529  
Priority Number (s):  
IPC Classification: G06F17/30 ; G06F17/27  
EC Classification:  
Equivalents: CN1237738

**Abstract**

**PROBLEM TO BE SOLVED:** To provide a method for extracting the feature of contents which are described in a document without using a word dictionary and a high speed pseudo document retrieving system with high precision, where the method is used.

**SOLUTION:** The system is provided with a step for storing the probability of a character string existing in a text 150 in a text database to appear in the boundary of a word in the text 150 as an appearance probability file 152, a step for storing the times of appearance of the character string existing in the text 150 as an appearance time file 153, a step for extracting a feature character string from the text which is designated by a user through the use of the file 152 and a step for counting the times of the appearance of the feature character string in the text which is designated by the user. Then, a similarity degree as against the text designated by the user is calculated through the use of the file 153 and the number of appearance times in the text which is designated by the user.



Data supplied from the esp@cenet database - I2

W81

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平11-338883

(43)公開日 平成11年(1999)12月10日

(51)Int.Cl.<sup>6</sup>

識別記号

F I

G 0 6 F 17/30  
17/27

G 0 6 F 15/401  
15/38  
15/40

3 1 0 A  
E  
3 7 0 A

審査請求 未請求 請求項の数11 O L (全 40 頁)

(21)出願番号 特願平10-148721

(22)出願日 平成10年(1998) 5 月29日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目 6 番地

(72)発明者 松林 忠孝

神奈川県横浜市都筑区加賀原二丁目 2 番

株式会社日立製作所システム開発本部内

(72)発明者 多田 勝己

神奈川県横浜市都筑区加賀原二丁目 2 番

株式会社日立製作所システム開発本部内

(72)発明者 岡本 卓哉

神奈川県横浜市都筑区加賀原二丁目 2 番

株式会社日立製作所システム開発本部内

(74)代理人 弁理士 小川 勝男

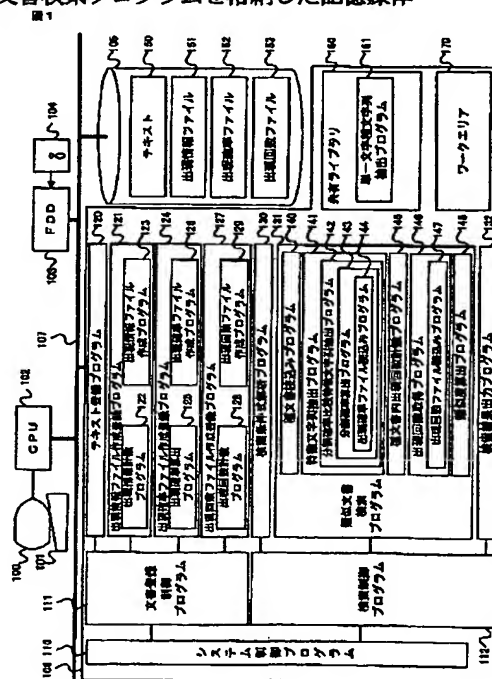
最終頁に続く

(54)【発明の名称】 特徴文字列抽出方法および装置とこれを用いた類似文書検索方法および装置並びに特徴文字列抽出プログラムを格納した記憶媒体および類似文書検索プログラムを格納した記憶媒体

(57)【要約】

【課題】本発明の課題は、単語辞書を用いずに文書に記述された内容の特徴を抽出する方法と、この方法を用いて、高速で高精度な類似文書検索システムを提供することである。

【解決手段】テキストデータベース中のテキスト150に存在する文字列のそのテキスト150における単語の境界に出現する確率を出現確率ファイル152として格納するステップと、テキスト150に存在する文字列の出現回数を出現回数ファイル153として格納するステップと、出現確率ファイル152を用いてユーザが指定したテキストから特徴文字列を抽出するステップと、ユーザが指定したテキストにおける特徴文字列の出現回数を計数するステップとを有し、出現回数ファイル153とユーザが指定したテキストにおける出現回数を用いてユーザが指定したテキストに対する類似度を算出する。



**【特許請求の範囲】**

【請求項1】テキストを含む文書から特徴を表す文字列（特徴文字列と呼ぶ）を抽出する特徴文字列抽出方法において、

単語間の区切れ目を境界として単語の候補となる文字列を上記テキストから抽出する文字列抽出ステップと、  
上記文字列抽出ステップで抽出された文字列中の長さが  $n$ （ $n$ は1以上の整数）の連続する文字列（ $n$ -gramと呼ぶ）に関するテキストデータベース内での出現回数を参照し、該出現回数が最大の $n$ -gramを特徴文字列として抽出する特徴 $n$ -gram抽出ステップとを有することを特徴とした特徴文字列抽出方法。

【請求項2】請求項1記載の特徴文字列抽出方法における前記文字列抽出ステップとして、  
所定の文字種の変わり目を境界としてテキストから単語の候補となる文字列を抽出する文字列抽出ステップを有することを特徴とした特徴文字列抽出方法。

【請求項3】請求項1記載の特徴文字列抽出方法における前記特徴 $n$ -gram抽出ステップとして、  
前記文字列抽出ステップで単語の候補として抽出された文字列から $m$ -gram（ $m$ は1以上の整数）と $n$ -gram（ $n$ は1以上の整数）を抽出し、

該 $m$ -gramと該 $n$ -gramに関するテキストデータベース内での出現回数を参照し、両者のうち出現回数の多い方を特徴文字列として抽出する特徴 $n$ -gram抽出ステップを有することを特徴とした特徴文字列抽出方法。

【請求項4】請求項1記載の特徴文字列抽出方法において、テキストデータベースへの文書登録時の処理として、

テキストから単語の区切れ目を抽出し、これを境界として単語の候補となる文字列を抽出する登録用文字列抽出ステップと、

上記登録用文字列抽出ステップで抽出された文字列（抽出文字列と呼ぶ）に関し、テキストデータベース中での出現回数を計数し、テキストデータベース中の全ての抽出文字列の出現回数に対する割合から出現確率を算出する出現確率算出ステップを有するとともに、

前記特徴 $n$ -gram抽出ステップにおいて、出現回数の代わりに該当する出現確率を参照し、前記文字列抽出ステップで抽出された文字列中の $n$ -gramの出現確率を参照し、該出現確率が最大の $n$ -gramを特徴文字列として抽出する特徴 $n$ -gram抽出ステップとを有することを特徴とした特徴文字列抽出方法。

【請求項5】文字情報をコードデータとして蓄積したテキストデータベースを対象として、ユーザが指定した文章あるいは文書（以後、まとめて文書と呼ぶ）と類似する文書を検索する類似文書検索方法において、  
ユーザが指定した文書のテキスト（指定テキストと呼ぶ）から、単語間の区切れ目を抽出し、これを境界として単語の候補となる文字列を抽出する文字列抽出ステッ

プと、

上記文字列抽出ステップで抽出された文字列の中から、長さが  $n$ （ $n$ は1以上の整数）の連続する文字列（ $n$ -gramと呼ぶ）に関するテキストデータベース内での出現回数を参照し、該出現回数が最大の $n$ -gramを特徴文字列として抽出する特徴 $n$ -gram抽出ステップと、

上記特徴 $n$ -gram抽出ステップで抽出された特徴文字列に対して、指定テキスト内の出現回数を計数する指定テキスト内出現回数計数ステップと、

上記特徴 $n$ -gram抽出ステップで抽出された特徴文字列に対して、テキストデータベース内の各文書における出現回数を取得するテキストデータベース内出現回数取得ステップと、

上記指定テキスト内出現回数計数ステップで計数した該特徴文字列の指定テキスト内の出現回数と、上記テキストデータベース内出現回数取得ステップで取得した該特徴文字列のテキストデータベース内の各文書における出現回数を用いて、指定テキストとテキストデータベース内の各文書の類似度を算出する類似度算出ステップと、  
上記類似度算出ステップで算出されたテキストデータベース内の各文書の指定テキストに対する類似度を、検索結果として出力する検索結果出力ステップとを有することを特徴とした類似文書検索方法。

【請求項6】請求項5記載の類似文書検索方法において、テキストデータベースへの文書登録処理として、  
テキストから単語の区切れ目を抽出し、これを境界として単語の候補となる文字列を抽出する登録用文字列抽出ステップと、

上記登録用文字列抽出ステップで抽出された文字列から、長さが1から該文字列自体の長さ $m$ までの全ての $n$ -gramを抽出し、該登録文書の識別番号と該登録文書のテキストにおける出現回数を組みとして、これを該当する出現回数ファイルへ格納する出現回数ファイル作成ステップを有するとともに、

前記テキストデータベース内出現回数取得ステップとして、

前記特徴 $n$ -gram抽出ステップで抽出された特徴文字列に対し、該当する上記出現回数ファイルを参照して、該特徴文字列のテキストデータベース内の各文書における該特徴文字列の出現回数を取得するテキストデータベース内出現回数取得ステップを有する類似文書検索方法。

【請求項7】請求項6記載の類似文書検索方法における前記文字列抽出ステップとして、

所定の文字種の変わり目を境界としてテキストから単語の候補となる文字列を抽出する文字列抽出ステップを有することを特徴とした類似文書検索方法。

【請求項8】テキストを含む文書の特徴を表す文字列（特徴文字列と呼ぶ）を抽出する特徴文字列抽出装置において、

単語間の区切れ目を境界として単語の候補となる文字列

を上記テキストから抽出する文字列抽出手段と、上記文字列抽出装置で抽出された文字列中の長さが  $n$  ( $n$ は1以上の整数)の連続する文字列( $n$ -gramと呼ぶ)に関するテキストデータベース内での出現回数を参照し、該出現回数が最大の $n$ -gramを特徴文字列として抽出する特徴 $n$ -gram抽出手段とを備えたことを特徴とした特徴文字列抽出装置。

【請求項9】文字情報をコードデータとして蓄積したテキストデータベースを対象として、ユーザが指定した文章あるいは文書(以後、まとめて文書と呼ぶ)と類似する文書を検索する類似文書検索装置において、ユーザが指定した文書のテキスト(指定テキストと呼ぶ)から、単語間の区切れ目を検出し、これを境界として単語の候補となる文字列を抽出する文字列抽出手段と、

上記文字列抽出手段で抽出された文字列の中から、長さが  $n$  ( $n$ は1以上の整数)の連続する文字列( $n$ -gramと呼ぶ)に関するテキストデータベース内での出現回数を参照し、該出現回数が最大の $n$ -gramを特徴文字列として抽出する特徴 $n$ -gram抽出手段と、

上記特徴 $n$ -gram抽出手段で抽出された特徴文字列に対して、指定テキスト内の出現回数を計数する指定テキスト内出現回数計数手段と、

上記特徴 $n$ -gram抽出手段で抽出された特徴文字列に対して、テキストデータベース内の各文書における出現回数を取得するテキストデータベース内出現回数取得手段と、

上記指定テキスト内出現回数計数ステップで計数した該特徴文字列の指定テキスト内の出現回数と、上記テキストデータベース内出現回数取得手段で取得した該特徴文字列のテキストデータベース内の各文書における出現回数を用いて、指定テキストとテキストデータベース内の各文書の類似度を算出する類似度算出手段と、上記類似度算出手段で算出したテキストデータベース内の各文書の指定テキストに対する類似度を、検索結果として出力する検索結果出力手段とを備えたことを特徴とした類似文書検索方法。

【請求項10】テキストを含む文書の特徴を表す文字列(特徴文字列と呼ぶ)を抽出する特徴文字抽出プログラムを格納する記憶媒体において、

単語間の区切れ目を境界として単語の候補となる文字列を上記テキストから抽出する文字列抽出ステップと、

上記文字列抽出ステップで抽出された文字列中の長さが  $n$  ( $n$ は1以上の整数)の連続する文字列( $n$ -gramと呼ぶ)に関するテキストデータベース内での出現回数を参照し、該出現回数が最大の $n$ -gramを特徴文字列として抽出する特徴 $n$ -gram抽出ステップとを有する特徴文字列抽出プログラムを格納することを特徴とした記憶媒体。

【請求項11】文字情報をコードデータとして蓄積したテキストデータベースを対象として、ユーザが指定した

文章あるいは文書(以後、まとめて文書と呼ぶ)と類似する文書を検索する類似文書検索プログラムを格納する記憶媒体において、

ユーザが指定した文書のテキスト(指定テキストと呼ぶ)から、単語間の区切れ目を検出し、これを境界として単語の候補となる文字列を抽出する文字列抽出ステップと、

上記文字列抽出ステップで抽出された文字列の中から、長さが  $n$  ( $n$ は1以上の整数)の連続する文字列( $n$ -gramと呼ぶ)に関するテキストデータベース内での出現回数を参照し、該出現回数が最大の $n$ -gramを特徴文字列として抽出する特徴 $n$ -gram抽出ステップと、

上記特徴 $n$ -gram抽出ステップで抽出された特徴文字列に対して、指定テキスト内の出現回数を計数する指定テキスト内出現回数計数ステップと、

上記特徴 $n$ -gram抽出ステップで抽出された特徴文字列に対して、テキストデータベース内の各文書における出現回数を取得するテキストデータベース内出現回数取得ステップと、

上記指定テキスト内出現回数計数ステップで計数した該特徴文字列の指定テキスト内の出現回数と、上記テキストデータベース内出現回数取得ステップで取得した該特徴文字列のテキストデータベース内の各文書における出現回数を用いて、指定テキストとテキストデータベース内の各文書の類似度を算出する類似度算出ステップと、上記類似度算出ステップで算出されたテキストデータベース内の各文書の指定テキストに対する類似度を、検索結果として出力する検索結果出力ステップを有する類似文書検索プログラムを格納することを特徴とした記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、文書に記述された内容の特徴を表す文字列を抽出する方法および装置並びに文字列抽出プログラムを格納した記憶媒体と、この方法および装置を用いて、ユーザが指定した文書に記述されている内容と類似する内容を含む文書を文書データベースの中から検索する方法および装置並びに検索プログラムを格納した記憶媒体に関する。

【0002】

【従来の技術】近年、パーソナルコンピュータやインターネット等の普及に伴い、電子化文書が爆発的に増加しており、今後も加速度的に増大していくものと予想される。このような状況において、ユーザが所望する情報を含んだ文書を高速かつ効率的に検索したいという要求が高まってきている。

【0003】このような要求に応える技術として全文検索がある。全文検索では、検索対象文書をテキストとして計算機システムに登録してデータベース化し、この中からユーザが指定した検索文字列(以下、検索タームと

呼ぶ)を含む文書を検索する。このように全文検索では、文書中の文字列そのものを対象として検索を行うため、予めキーワードを付与し、このキーワードを手掛りに検索する従来のキーワード検索システムとは異なり、どんな言葉でも検索ができるという特長がある。

【0004】しかし、ユーザが所望する情報を含んだ文書を的確に検索するためには、ユーザの検索意図を正確に表わす複雑な検索条件式を作成し、入力する必要がある。これは、情報検索の専門家でない一般のユーザにとっては容易なことではない。

【0005】この繁雑さを解消するために、ユーザが自分の所望する内容を含んだ文書(以下、種文書と呼ぶ)を例示し、その文書と類似する文書を検索する類似文書検索技術が注目されている。

【0006】類似文書検索の方法としては、例えば、「特開平8-335222号公報」に、形態素解析により種文書中に含まれる単語を抽出し、これを用いて類似文書を検索する技術(以下、従来技術1と呼ぶ)が開示されている。

【0007】従来技術1では、形態素解析により種分書中に含まれる単語を抽出し、この単語を含む文書を類似文書として検索する。例えば、文書1「・・・携帯電話の使用時のマナーが問題になる。・・・」を種文書とする場合、形態素解析により単語辞書を参照して、「携帯電話」「マナー」「問題」等の単語を抽出する。この結果、「携帯電話」を含む文書2「・・・電車内での携帯電話の使用は禁止されている・・・」を類似文書として検索することができる。

【0008】しかし、従来技術1では、単語の抽出に単語辞書を用いるため、次のような2つの問題がある。

【0009】まず、単語辞書に掲載されていない単語が文書の本質的な内容(以下、中心概念と呼ぶ)を表わす場合、この単語が種文書から検索用の単語として抽出されないため、他の単語によって類似検索が行われたとしても、文書の中心概念が正確に検索できない恐れがある。すなわち、ユーザが所望する情報が新語で表されるような場合、これが単語辞書に含まれていないと、目的とする中心概念からずれた文書が検索されてしまうという問題がある。

【0010】次に、ユーザが所望する情報を表わす言葉が単語辞書に掲載されている場合でも、単語の抽出の仕方によっては検索の対象とする中心概念がずれてしまうという問題がある。例えば、上記の文書1「・・・携帯電話の使用のマナーが問題になる。・・・」という種文書からは、「携帯電話」「マナー」「問題」等の単語が抽出される。しかし、「電話」という単語が抽出されないため文書3「・・・電話での話し方について注意された。・・・」という文書の類似度が低く算出されてしまう恐れがある。

【0011】これらは、全て単語辞書を用いて検索用の

単語を抽出する方法を用いていることに起因する。

【0012】以上が従来技術1の問題点である。

【0013】この問題を解決するために、「特願平9-309078号」で、単語辞書を用いずに、種文書中から漢字やカタカナ等の文字種別に連続するn文字の文字列(以下、n-gramと呼ぶ)を漢字やカタカナ等の文字種別に機械的に抽出し、これを用いて類似文書を検索する技術(以下、従来技術2と呼ぶ)を提案した。

【0014】従来技術2では、文字種別にn-gramの抽出方法を変え、意味のまとまりをもったn-gram(以下、特徴文字列と呼ぶ)を抽出する。例えば、漢字で構成される文字列(以下、漢字文字列と呼ぶ)からは機械的に2-gramを抽出し、カタカナで構成される文字列(以下、カタカナ文字列と呼ぶ)からは、カタカナで構成される最長の文字列(以下、カタカナ最長文字列と呼ぶ)、すなわちカタカナ文字列そのものを抽出する。この場合、上記の文書1「・・・携帯電話の使用のマナーが問題になる。・・・」という種文書からは、「携帯」「帯電」「電話」「使用」「マナー」「問題」等という特徴文字列が抽出される。すなわち、「電話」という文字列も漏れなく抽出されるため、従来技術1では低い類似度が算出されてしまう文書3「・・・電話での話し方について注意された。・・・」についても正しく類似度が算出されるようになる。

【0015】しかし、従来技術2では、複合語を構成する可能性のある漢字文字列等からは、単語間にまたがるn-gramも抽出する可能性がある。このため、これを検索に用いると、内容の類似しない文書に対してまでも類似度が算出され、この結果、関連のない文書が類似文書として検索されるという問題が生じる。例えば、上記の文書1「・・・携帯電話の使用のマナーが問題になる。・・・」という種文書から抽出された「帯電」という特徴文字列により類似度が算出され、文書4「・・・電荷の帯電を防ぐために、接地しなくてはならない。・・・」という文書が類似文書として誤って検索されてしまうという問題がある。

【0016】この問題を解決するための技術として、「情報処理学会論文誌 pp.2286~2297, Vol.38, No.11, Nov.1997」に、1-gramの統計情報を用いて特徴文字列を抽出する技術(以下、従来技術3と呼ぶ)が提案されている。

【0017】従来技術3では、文書登録時に登録文書中に出現する各1-gramについて、単語の先頭である確率(以下、先頭確率と呼ぶ)と末尾である確率(以下、末尾確率と呼ぶ)を算出しておく。ここでは、単語を、漢字やカタカナ等の文字種境界で区切られ、単一の文字種で構成される文字列(以下、単一文字種文字列と呼ぶ)とし、文字種境界の直後に位置する1-gramを単語の先頭にある1-gramとし、文字種境界の直前に位置する1-gramを単語の末尾にある1-gramとしている。

【0018】例えば、上記の文書1「・・・携帯電話の使用のマナーが問題になる・・・」から文字種境界で抽出した“使用”という漢字文字列では、“使”が単語の先頭にある1-gramで、“用”が単語の末尾にある1-gramとなる。

【0019】類似文書検索時には、まず指定された種文書から単一文字種文字列を抽出する。次に、単一文字種文字列内の連続する2個の1-gramにおける前方の1-gramの末尾確率と後方の1-gramの先頭確率から、これらの1-gram間で単一文字種文字列が分割される確率（以下、分割確率と呼ぶ）を算出し、この値が所定の値（以下、分割閾値と呼ぶ）を越えている場合には、そこで単一文字種文字列を分割するという処理を行う。

【0020】以下、分割閾値を0.050として、従来技術3の具体的な処理方法を説明する。

【0021】まず、文書登録時には全登録対象文書中に出現する各1-gramについて、出現回数、単語の先頭に出現する回数（以下、先頭回数と呼ぶ）および末尾に出現する回数（以下、末尾回数と呼ぶ）を計数し、出現情報ファイルに格納する。例えば、上記の文書1では“携”の出現回数は1回、先頭回数は1回および末尾回数は0回という出現情報が得られる。図2に出現情報ファイルの例を示す。

【0022】その後、上記出現情報ファイルを参照し、各1-gramについて、それぞれ先頭確率と末尾確率を算出し、出現確率ファイルに格納する。例えば、1-gram“携”の先頭確率は $768 / 4,740 = 0.16$ 、末尾確率は $92 / 4,740 = 0.10$ となる。図3に出現確率ファイルの例を示す。

【0023】次に、単一文字種文字列「携帯電話」を例として、従来技術3の文書検索方法を説明する。

【0024】まず、単一文字種文字列「携帯電話」の中から1-gramの二つの組として、（“携”，“帯”）、（“帯”，“電”）および（“電”，“話”）の3個を抽出する。次に、各1-gramの組において、前方の1-gramの末尾確率と後方の1-gramの先頭確率を、登録時に作成した出現確率ファイルから取得し、分割確率を算出する。

【0025】図4に、「携帯電話」から抽出した3個の1-gramの組における分割確率の算出過程を示す。本例では、（“携”，“帯”）、（“帯”，“電”）および（“電”，“話”）の分割確率として、それぞれ0.011、0.054および0.005が算出され、これらの分割確率のうち、（“帯”，“電”）の0.054が分割閾値0.050より大きいので、“帯”と“電”の間で分割される。一方、（“携”，“帯”）および（“電”，“話”）の分割確率はそれぞれ0.011および0.005であり、これらは分割閾値0.050より小さいので、これらの1-gram間では分割されない。その結果、「携帯電話」が“帯”と“電”の間で分割され、「携帯」と「電話」の2個の特徴文字列が

抽出されることになる。

【0026】以上が、従来技術3の具体的な処理方法である。このように従来技術3では、1-gramの統計情報を用いて特徴文字列を抽出することにより、単語間にまたがる不適切な特徴文字列を抽出しないようにして、内容の類似しない文書が検索されることのないように配慮している。

【0027】しかし、従来技術3では、分割確率の絶対値で分割の可否を判断するため、単語としての特徴文字列の抽出精度が低いという問題がある。例えば、単一文字種文字列「帯電」に対しては、1-gramの組（“帯”，“電”）が抽出され、この1-gram間の分割確率として0.054が算出される。

【0028】この値は分割閾値0.050より大きいので、「帯電」が「帯」と「電」のように誤って分割（以下、誤分割と呼ぶ）されてしまい、不適切な2個の特徴文字列が抽出されてしまう。この結果、「帯（おび）」に関係のある文書等も類似文書として検索されてしまい、検索ノイズが混入して、目的とする中心概念がずれた文書が類似文書として検索されてしまうという問題がある。

【0029】

【発明が解決しようとする課題】以上述べたように、従来技術1のように単語辞書を用いて単語を抽出する方法では、単語辞書に掲載されていない単語が種文書の中心概念を表す場合には、中心概念からずれた文書が検索されてしまうという問題がある。

【0030】また、従来技術2のように単一文字種文字列から文字種別に、単純にn-gramを抽出する方法では、複合語を構成する可能性のある漢字文字列等から単語間にまたがるn-gramを抽出してしまうことにより、関連のない文書が類似文書として検索されてしまうという問題がある。

【0031】さらに、従来技術3のように、1-gramの統計情報を用いて分割確率を算出し、この値の絶対値で分割の可否を判断する方法においても、単語としての特徴文字列の抽出精度が低いので、検索ノイズが混入し、目的とする中心概念がずれた文書が類似文書として検索されてしまうという問題がある。

【0032】こうした従来技術の問題に対し、本発明では、誤分割が少なくなるように特徴文字列を抽出する方法および装置を提供することを目的とする。

【0033】また、誤分割が少なくなるように特徴文字列を抽出することにより、検索ノイズを少なくすることで中心概念のずれを低減した類似文書検索が行える方法および装置を提供することを目的とする。

【0034】

【課題を解決するための手段】上記課題を解決するために、本発明による特徴文字列抽出方法では、以下に示すステップからなる処理により、種文書から特徴文字列の抽出を行なう。



【0035】すなわち、本発明による特徴文字列抽出方法では、文書の登録処理として、

(ステップ1) 登録対象文書を読み込む文書読み込みステップ、

(ステップ2) 上記文書読み込みステップで読み込んだ登録対象文書中の文字列を、漢字やカタカナ等の文字種境界で分割し、単一文字種文字列として抽出する単一文字種文字列抽出ステップ、

(ステップ3) 上記単一文字種文字列抽出ステップで抽出された単一文字種文字列に対して、その文字種を判定し、漢字やカタカナならば予め定められた長さのn-gramについて登録文書における出現回数、単語の先頭に出現する回数(以下、先頭回数と呼ぶ)と末尾に出現する回数(以下、末尾回数と呼ぶ)、およびn-gramそのものが単語として出現する回数(以下、単独回数と呼ぶ)を計数する出現情報計数ステップ、

(ステップ4) 上記出現情報計数ステップで計数されたn-gramの出現情報を、既にデータベースに登録されている文書に関する該n-gramの出現情報に加算することで、データベース全体の出現情報を算出し、該当する出現情報ファイルへ格納する出現情報ファイル作成登録ステップ、

(ステップ5) 上記出現情報計数ステップで出現情報が計数されたn-gramに関して、該当する出現情報ファイルからデータベース全体における出現情報を取得し、単語の先頭である確率(以下、先頭確率と呼ぶ)と末尾である確率(以下、末尾確率と呼ぶ)およびn-gramそのものが単語として出現する確率(以下、単独確率と呼ぶ)を算出し、該当する出現確率ファイルに格納する出現確率ファイル作成登録ステップ、

(ステップ6) 上記単一文字種文字列抽出ステップで抽出された単一文字種文字列から、予め定められた長さのn-gramを抽出し、登録対象文書中における出現回数を計数する出現回数計数ステップ、

(ステップ7) 上記出現回数計数ステップで計数された出現回数を該当する出現回数ファイルに格納する出現回数ファイル作成登録ステップ、を有し、種文書から特徴文字列を抽出する処理として、

(ステップ8) 種文書を読み込む種文書読み込みステップ、

(ステップ9) 上記種文書読み込みステップにおいて読み込まれた種文書中の文字列を文字種境界で分割し、単一文字種文字列として抽出する検索用単一文字種文字列抽出ステップ、

(ステップ10) 上記検索用単一文字種文字列抽出ステップで抽出された単一文字種文字列に関して、その文字種を判定し、漢字やカタカナならば、前記出現確率ファイルを読み込み、単一文字種文字列の先頭からi文字目までの文字列の単独確率、(i+1)文字目までの文字列の単独確率、(i+1)文字目の文字の先頭確率および(i+2)文

字目の文字の先頭確率を取得し、i文字目で単一文字種文字列が分割される確率(以下、分割確率と呼ぶ)をi文字目までの文字列の単独確率と(i+1)文字目の文字の先頭確率の積として算出し、(i+1)文字目での分割確率を、(i+1)文字目までの文字列の単独確率と(i+2)文字目の文字の先頭確率の積として算出し、これらのi文字目と(i+1)文字目の分割確率を比較して、値の大きい方を単一文字種文字列が分割される点(以下、分割点と呼ぶ)とし、先頭から分割点までの文字列を特徴文字列として抽出し、漢字やカタカナ以外ならば、単一文字種文字列そのものを特徴文字列として抽出し、抽出された特徴文字列を除外した残りの文字列に対して、同様の処理を繰り返すことによって特徴文字列を抽出する特徴文字列抽出ステップを有する。

【0036】また、前述の課題を解決するために、本発明による類似文書検索方法では、上記ステップからなる処理により、種文書と類似する文書を検索するための特徴文字列を抽出し、これを用いて類似文書検索を行う。

【0037】すなわち、本発明による類似文書検索方法では、文書の登録処理として、

(ステップ1) 登録対象文書を読み込む文書読み込みステップ、

(ステップ2) 上記文書読み込みステップで読み込んだ登録対象文書中の文字列を、漢字やカタカナ等の文字種境界で分割し、単一文字種文字列として抽出する単一文字種文字列抽出ステップ、

(ステップ3) 上記単一文字種文字列抽出ステップで抽出された単一文字種文字列に対して、その文字種を判定し、漢字やカタカナならば予め定められた長さのn-gramについて登録文書における出現回数、単語の先頭に出現する回数(以下、先頭回数と呼ぶ)と末尾に出現する回数(以下、末尾回数と呼ぶ)、およびn-gramそのものが単語として出現する回数(以下、単独回数と呼ぶ)を計数する出現情報計数ステップ、

(ステップ4) 上記出現情報計数ステップで計数されたn-gramの出現情報を、既にデータベースに登録されている文書に関する該n-gramの出現情報に加算することで、データベース全体の出現情報を算出し、該当する出現情報ファイルへ格納する出現情報ファイル作成登録ステップ、

(ステップ5) 上記出現情報計数ステップで出現情報が計数されたn-gramに関して、該当する出現情報ファイルからデータベース全体における出現情報を取得し、単語の先頭である確率(以下、先頭確率と呼ぶ)と末尾である確率(以下、末尾確率と呼ぶ)およびn-gramそのものが単語として出現する確率(以下、単独確率と呼ぶ)を算出し、該当する出現確率ファイルに格納する出現確率ファイル作成登録ステップ、

(ステップ6) 上記単一文字種文字列抽出ステップで抽出された単一文字種文字列から、予め定められた長さの

n-gramを抽出し、登録対象文書中における出現回数を計数する出現回数計数ステップ、

(ステップ7) 上記出現回数計数ステップで計数された出現回数を該当する出現回数ファイルに格納する出現回数ファイル作成登録ステップ、を有し、種文書に類似する文書の検索処理として、

(ステップ8) 種文書を読み込む種文書読み込みステップ、

(ステップ9) 上記種文書読み込みステップにおいて読み込まれた種文書中の文字列を文字種境界で分割し、単一文字種文字列として抽出する検索用単一文字種文字列抽出ステップ、

(ステップ10) 上記検索用単一文字種文字列抽出ステップで抽出された単一文字種文字列に関して、その文字種を判定し、漢字やカタカナならば、前記出現確率ファイルを読み込み、単一文字種文字列の先頭から i 文字目までの文字列の単独確率、(i+1)文字目までの文字列の単独確率、(i+1)文字目の文字の先頭確率および(i+2)文字目の文字の先頭確率を取得し、i文字目で単一文字種文字列が分割される確率(以下、分割確率と呼ぶ)を i 文字目までの文字列の単独確率と(i+1)文字目の文字の先頭確率の積として算出し、(i+1)文字目での分割確率を、(i+1)文字目までの文字列の単独確率と(i+2)文字目の文字の先頭確率の積として算出し、これらの i 文字目と(i+1)文字目の分割確率を比較して、値の大きい方を単一文字種文字列が分割される点(以下、分割点と呼ぶ)とし、先頭から分割点までの文字列を特徴文字列として抽出し、漢字やカタカナ以外ならば、単一文字種文字列そのものを特徴文字列として抽出し、抽出された特徴文字列を除外した残りの文字列に対して、同様の処理を繰り返すことによって特徴文字列を抽出する特徴文字列抽出ステップ、

(ステップ11) 上記特徴文字列抽出ステップで抽出された全ての特徴文字列に対して、種文書内における出現回数を計数する種文書内出現回数計数ステップ、

(ステップ12) 上記特徴文字列抽出ステップで抽出された全ての特徴文字列に対して、前記出現回数ファイルを読み込み、データベース内の各文書における該当特徴文字列の出現回数を取得するデータベース内出現回数取得ステップ、

(ステップ13) 上記特徴文字列抽出ステップで抽出された特徴文字列に対し、上記種文書内出現回数計数ステップで計数された種文書内の出現回数と、上記データベース内出現回数取得ステップで取得されたデータベース内の各文書における出現回数を用いて、予め定められた算出式に基づいて種文書とデータベース内の各文書との類似度を算出する類似度算出ステップ、

(ステップ14) 上記類似度算出ステップで算出された類似度に基づいて、検索結果を出力する検索結果出力ステップを有する。

【0038】上記文書検索方法を用いた本発明の原理を、以下に説明する。

【0039】本発明では、文書を登録する際に、(ステップ1)～(ステップ7)を実行する。

【0040】まず、文書読み込みステップ(ステップ1)で登録対象となる文書を読み込む。次に、単一文字種文字列抽出ステップ(ステップ2)において、上記文書読み込みステップ(ステップ1)で読み込まれた登録対象文書中の文字列を、漢字やカタカナ等の文字種境界で分割し、単一文字種からなる文字列を抽出する。例えば、前述の文書2「・・・」。電車内での携帯電話の使用は禁止されている。・・・」という文書からは、「電車内」「での」「携帯電話」「の」「使用」「は」「禁止」「されている」等の単一文字種文字列が抽出される。

【0041】次に、出現情報計数ステップ(ステップ3)において、単一文字種文字列抽出ステップ(ステップ2)で抽出された上記各単一文字種文字列について、その文字種を判定し、漢字やカタカナならば予め定められた長さ n の n-gram の登録対象文書中の出現回数、先頭回数、末尾回数および単独回数を計数する。例えば、漢字文字列とカタカナ文字列から1-gram および2-gram の出現回数、先頭回数および末尾回数を計数するものと定められている場合には、上記単一文字種文字列抽出ステップ(ステップ2)で抽出された単一文字種文字列について、“携”の出現回数は1回、そのうち先頭回数は1回、末尾回数は0回、単独回数は0回であり、“携帯”の出現回数は1回、そのうち先頭回数は1回、末尾回数は0回、単独回数は0回と計数される。

【0042】次に、出現情報ファイル作成登録ステップ(ステップ4)において、先に出現情報計数ステップ(ステップ3)で抽出された n-gram の出現情報を、既にデータベースに登録されている文書に関する出現情報に加算し、累積情報としての出現情報を該当する出現情報ファイルへ格納する。図5に出現情報ファイルの例を示す。本図に示した出現情報ファイルは、上記出現情報計数ステップ(ステップ3)において抽出された出現情報を格納した場合の例である。本図に示した出現情報ファイルは、前述の1-gram “携” に関しては、出現回数4,740回、先頭回数768回、末尾回数492回、および単独回数42回という情報を格納し、2-gram “携帯” に関しては、出現回数462回、先頭回数419回、末尾回数52回、および単独回数48回という情報を格納していることを表わす。

【0043】次に、出現確率ファイル作成登録ステップ(ステップ5)において、出現情報ファイル作成登録ステップ(ステップ4)で出現情報が格納された n-gram に対して、それぞれ出現確率を算出し、該当する出現確率ファイルに格納する。例えば、図5に示すように、1-gram “携” に関しては、出現回数4,740回、先頭回数768回、末尾回数492回、および単独回数42回であることから、先頭確率は  $768 / 4,740 = 0.16$ 、末尾確率は  $492 /$



4,740 = 0.10、単独確率は  $42 / 4,740 = 0.01$  と計算される。図6に出現確率ファイルの例を示す。本図に示した出現確率ファイルは、上記出現情報計数ステップ（ステップ3）において抽出された出現確率を格納した場合の例であり、前述の1-gram “携” に関しては、先頭確率0.16、末尾確率0.10、および単独確率0.01という情報が格納され、2-gram “携帯” に関しては、先頭確率0.90、末尾確率0.11、および単独確率0.10という情報が格納されていることを表わす。

【0044】次に、出現回数計数ステップ（ステップ6）において、単一文字種文字列抽出ステップ（ステップ2）で抽出された全ての単一文字種文字列から、予め定められた長さのn-gramを抽出し、登録対象文書中における出現回数を計数する。そして、出現回数ファイル作成登録ステップ（ステップ7）において、上記出現回数計数ステップ（ステップ6）で抽出された各n-gramの出現回数を該当する出現回数ファイルに格納する。

【0045】図24に、前述の文書2「・・・電車内での携帯電話の使用は禁止されている。・・・」を例に、出現回数ファイル作成処理の手順を示す。

【0046】まず、単一文字種文字列抽出ステップ（ステップ2）で登録対象文書である文書2から全ての単一文字種文字列を抽出する。

【0047】次に、出現回数計数ステップ（ステップ6）で、上記単一文字種文字列抽出ステップ（ステップ2）で抽出された全ての単一文字種文字列から予め定められた長さのn-gramを抽出し、登録対象文書内の出現回数を計数する。本図に示した例では、単一文字種文字列から長さが3のn-gramまでを抽出するものとし、単一文字種文字列2404に含まれる「電車内」から、長さが1の“電”、“車”、“内”、長さが2の“電車”、“車内”、および長さが3の“電車内”が抽出され、文書2における出現回数が計数される。この結果、“電”は文書2の中に2回出現し、“車”は文書2の中に1回出現しているというように計数される。

【0048】そして、出現回数ファイル作成登録ステップ（ステップ7）で、出現回数計数ステップ（ステップ6）で抽出された各n-gramの出現回数を該当する出現回数ファイルに格納する。この結果、文書2からは、1-gram “電”（2, 2），“車”（2, 1），“内”（2, 1）、2-gram “電車”（2, 1），“車内”（2, 1）、3-gram “電車内”（2, 1）のように各n-gramの登録対象文書の識別番号と出現回数が組みとして格納される。ここで、“電車”（2, 1）は、2-gram “電車”が文書番号2の文書に、1回出現するということを示している。

【0049】検索時には、（ステップ8）～（ステップ14）を実行する。

【0050】まず、種文書読み込みステップ（ステップ8）において、種文書として文書1を読み込む。次

に、検索用単一文字種文字列抽出ステップ（ステップ9）において、上記種文書読み込みステップ（ステップ8）で読み込まれた種文書（文書1）中の文字列を文字種境界で分割し、単一文字種文字列を抽出する。

【0051】次に、特徴文字列抽出ステップ（ステップ10）において、上記検索用単一文字種文字列抽出ステップ（ステップ9）で抽出された単一文字種文字列について、その文字種を判定する。

【0052】この文字種が、漢字やカタカナならば、前述した出現確率ファイルを読み込み、単一文字種文字列の先頭からi文字目までの文字列の単独確率、(i+1)文字目までの文字列の単独確率、(i+1)文字目の文字の先頭確率および(i+2)文字目の文字の先頭確率を取得する。そして、i文字目での分割確率をi文字目までの文字列の単独確率と(i+1)文字目の文字の先頭確率の積として算出し、(i+1)文字目での分割確率を(i+1)文字目までの文字列の単独確率と(i+2)文字目の文字の先頭確率の積として算出する。そして、これらのi文字目と(i+1)文字目の分割確率を比較して、値の大きい方を分割点とし、先頭から該分割点までの文字列を特徴文字列として抽出する。

【0053】また、漢字やカタカナでなければ、単一文字種文字列そのものを特徴文字列として抽出し、以下、同様の処理を繰り返すことによって、特徴文字列を抽出する。

【0054】図8に、文書1から抽出した単一文字種文字列「携帯電話」から特徴文字列を抽出する例を示す。まず、「携帯電話」における1文字目での分割確率は、「携」の単独確率0.01と「帯」の先頭確率0.11の積として0.001が算出され、2文字目での分割確率は、「携帯」の単独確率0.10と「電」の先頭確率0.36の積として0.036が算出される。次に、これらの分割確率を比較し、値の大きい方で単一文字種文字列を分割する。この場合、2文字目の分割確率0.036の方が大きいので、単一文字種文字列「携帯電話」は「携帯」と「電話」に分割される。

【0055】また、図9に、従来技術3では適切に分割されない単一文字種文字列「帯電」の例について、本発明の分割処理を示す。まず、「帯電」における1文字目での分割確率は、「帯」の単独確率0.01と「電」の単独確率0.01の積として0.0001と算出される。また、2文字目での分割確率、すなわち「帯電」が単一文字種文字列そのものとして出現する確率は、「帯電」の単独確率0.10と算出される。これらの値を比較して、値の大きい方で単一文字種文字列に分割される。この場合、「帯電」の単独確率0.10の方が大きいので、「帯電」は2文字目で分割されることになり、結果的に単一文字種文字列「帯電」は分割されず、一塊の文字列として抽出されることになる。

【0056】このように分割確率を比較して単一文字種

文字列を分割することにより、データベース中での実際の出現状況を正確に反映した単語分割が行なえるため、分割確率の絶対値で分割する前述した従来技術3に比べ、不適切な分割を大幅に削減することが可能になる。

【0057】次に、種文書内出現回数計数ステップ（ステップ11）において、上記特徴文字列抽出ステップ（ステップ10）で抽出された特徴文字列の種文書内における出現回数を計数する。

【0058】そして、データベース内出現回数取得ステップ（ステップ12）において、上記特徴文字列抽出ステップ（ステップ10）で抽出された特徴文字列に対して、前述した出現回数ファイルを参照し、データベース内の各文書における出現回数を得る。

【0059】そして、類似度算出ステップ（ステップ13）において、前記特徴文字列抽出ステップ（ステップ10）で抽出された特徴文字列に対して、上記種文書内出現回数計数ステップ（ステップ11）とデータベース内出現回数取得ステップ（ステップ12）で計数された種文書内における出現回数と、データベース内の各文書における出現回数を基に、類似度が算出される。

【0060】類似度の算出には、例えば、「特開平6-110948号公報」に開示されている以下に示す類似度算出式（1）を用いてもよい。

【0061】

【数式1】

$$S(i) = \frac{\sum_{j=1}^n U(j) \times R(j)}{\sqrt{\sum_{j=1}^n U(j)^2 \times \sum_{j=1}^n R(j)^2}} \quad \dots (1)$$

【0062】ここで、 $U(j)$ は種文書中の  $j$  番目の  $n$ -gram の正規化ウエイトを示し、各  $n$ -gram の種文書内出現回数から算出される。 $R(j)$ はデータベース中文書の  $j$  番目の  $n$ -gram の正規化ウエイトを示し、各  $n$ -gram のデータベース内の各文書における出現回数から算出される。正規化ウエイトとは、データベースにおける  $n$ -gram の出現偏りを表し、この値が大きい  $n$ -gram ほどある特定の文書に偏って出現することを意味する。この正規化ウエイトの算出方法については、「特開平6-110948号公報」で説明されているため、ここでは説明を省略する。また、 $n$ はデータベース中の全文書数を表わす。

【0063】この類似度算出式（1）を用いて、文書1が種文書として指定された場合の文書  $i$  の類似度  $S(i)$  を算出すると、次のようになる。

【0064】 $S(1) = 1.0$

$S(2) = 0.262$

$S(3) = 0.048$

$S(4) = 0.0$

この結果、検索結果出力ステップ（ステップ14）で、文書を類似度の降順に整列すると、文書1、文書2、お

よび文書3の順に表示されることになる。類似度が0の文書4は検索結果としては出力されない。

【0065】以上説明したように、本発明の特徴文字列抽出方法を用いた類似文書検索方法によれば、従来技術1のように単語辞書を用いることなく単一文字種文字列から文字列を機械的に抽出することができるため、どのような単語についても漏れなく検索に供することができ、種文書が表わす概念を正確に検索することが可能となる。

【0066】また、従来技術2のように単一文字種文字列から文字種別に、単純に  $n$ -gram を抽出するのではなく、統計情報を用いて意味のまとまった  $n$ -gram を抽出することにより、種文書が表わす概念をより正確に検索することが可能となる。

【0067】さらに、従来技術3のように分割確率の絶対値で分割するのではなく、分割確率を比較し、その値が大きい方で分割することにより、データベース中での実際の出現状況を正確に反映した単語分割が可能となり、不適切な単語分割を大幅に削減することが可能となる。そのため、従来技術3に比べ不適切な特徴文字列が検索に供されないため、種文書が表わす概念を適切に検索できるとともに、高速に類似文書を検索することができるようになる。

【0068】

【発明の実施の形態】以下、本発明の第一の実施例について図1を用いて説明する。

【0069】本発明を適用した類似文書検索システムの第一の実施例は、ディスプレイ100、キーボード101、中央演算処理装置（CPU）102、磁気ディスク装置105、フロッピーディスクドライブ（FDD）103、主メモリ106およびこれらを結ぶバス107から構成される。

【0070】磁気ディスク装置105には、テキスト150、出現情報ファイル151、出現確率ファイル152および出現回数ファイル153が格納される。FDD103を介してフロッピーディスク104に格納されている登録文書や種文書等の情報が、主メモリ106内に確保されるワークエリア170あるいは磁気ディスク装置105へ読み込まれる。

【0071】主メモリ106には、システム制御プログラム110、文書登録制御プログラム111、共有ライブラリ160、テキスト登録プログラム120、出現情報ファイル作成登録プログラム121、出現確率ファイル作成登録プログラム124、出現回数ファイル作成登録プログラム127、検索制御プログラム112、検索条件式解析プログラム130、類似文書検索プログラム131および検索結果出力プログラム132が格納されるとともにワークエリア170が確保される。これらのプログラムは、フロッピーディスクやCD-ROMなどの持ち運び可能な記憶媒体に格納され、ここから読み出

し磁気ディスク装置105へインストールする。本装置起動時に、システム制御プログラム110が起動し、これらのプログラムを磁気ディスク装置105から読み出し、主メモリ106へ格納する。

【0072】共有ライブラリ160は、単一文字種文字列抽出プログラム161で構成される。

【0073】出現情報ファイル作成登録プログラム121は、出現情報計数プログラム122と出現情報ファイル作成プログラム123で構成されるとともに、後述するように共有ライブラリ160から単一文字種文字列抽出プログラム161を呼び出す構成をとる。

【0074】出現確率ファイル作成登録プログラム124は、出現確率算出プログラム125と出現確率ファイル作成プログラム126で構成される。

【0075】出現回数ファイル作成登録プログラム127は、出現回数計数プログラム128と出現回数ファイル作成プログラム129で構成される。

【0076】類似文書検索プログラム131は、種文書読み込みプログラム140、特徴文字列抽出プログラム141、種文書内出現回数計数プログラム145、出現回数取得プログラム146および類似度算出プログラム148で構成されるとともに、後述するように共有ライブラリ160から単一文字種文字列抽出プログラム161を呼び出す構成をとる。

【0077】特徴文字列抽出プログラム141は、分割確率比較特徴文字列抽出プログラム142を呼び出す構成をとる。分割確率比較特徴文字列抽出プログラム142は、分割確率算出プログラム143を呼び出す構成をとる。分割確率算出プログラム143は出現確率ファイル読み込みプログラム144を呼び出す構成をとる。

【0078】出現回数取得プログラム146は、出現回数ファイル読み込みプログラム147を呼び出す構成をとる。

【0079】文書登録制御プログラム111および検索制御プログラム112は、ユーザによるキーボード101からの指示に応じてシステム制御プログラム110によって起動され、それぞれテキスト登録プログラム120、出現情報ファイル作成登録プログラム121、出現確率ファイル作成登録プログラム124および出現回数ファイル作成登録プログラム127の制御と、検索条件式解析プログラム130、類似文書検索プログラム131および検索結果出力プログラム132の制御を行なう。

【0080】以下、本実施例における類似文書検索システムの処理手順について説明する。

【0081】まず、システム制御プログラム110の処理手順について図10のPAD (Problem Analysis Diagram) 図を用いて説明する。

【0082】システム制御プログラム110では、まずステップ1000で、キーボード101から入力された

コマンドを解析する。

【0083】次に、ステップ1001で、この解析結果が登録実行のコマンドであると判定された場合には、ステップ1002で文書登録制御プログラム111を起動して、文書の登録を行なう。

【0084】またステップ1003で、検索実行のコマンドであると判定された場合には、ステップ1004で検索制御プログラム112を起動して、類似文書の検索を行なう。

【0085】以上が、システム制御プログラム110の処理手順である。

【0086】次に、図10に示したステップ1002でシステム制御プログラム110により起動される文書登録制御プログラム111の処理手順について、図11のPAD図を用いて説明する。

【0087】文書登録制御プログラム111では、まずステップ1100でテキスト登録プログラム120を起動し、FDD103に挿入されたフロッピディスク104から登録すべき文書のテキストデータをワークエリア170に読み込み、これをテキスト150として磁気ディスク装置105に格納する。テキストデータは、フロッピディスク104を用いて入力するだけに限らず、通信回線やCD-ROM装置 (図1には示していない) 等を用いて他の装置から入力するような構成を取ることも可能である。

【0088】次に、ステップ1101で出現情報ファイル作成登録プログラム121を起動し、ワークエリア170に格納されているテキスト150を読み出し、その中の各n-gramに対する出現情報ファイル151を作成し、磁気ディスク装置105に格納する。

【0089】次に、ステップ1102で出現確率ファイル作成登録プログラム124を起動し、ワークエリア170に格納されているテキスト150中の各n-gramに対する出現確率を算出し、該当する出現確率ファイル152として、磁気ディスク装置105へ格納する。

【0090】次に、ステップ1103で出現回数ファイル作成登録プログラム127を起動し、ワークエリア170に格納されているテキスト150を読み出し、その中の各文書における全てのn-gramに対する出現回数を計数し、該当する出現回数ファイル153として、磁気ディスク装置105へ格納する。

【0091】以上が、文書登録制御プログラム111の処理手順である。

【0092】次に、図11に示したステップ1101で文書登録制御プログラム111により起動される出現情報ファイル作成登録プログラム121の処理手順について、図12のPAD図を用いて説明する。

【0093】出現情報ファイル作成登録プログラム121では、まずステップ1200で単一文字種文字列抽出プログラム161を起動し、テキスト150の文字列を

文字種境界で分割することにより単一文字種文字列を抽出し、ワークエリア170に格納する。

【0094】次に、ステップ1201において、出現情報計数プログラム122を起動し、テキスト150における予め定められた長さのn-gramの出現回数と、ワークエリア170に格納されている単一文字種文字列の先頭回数、末尾回数および単独回数を計数し、同じくワークエリア170に格納する。

【0095】そして、ステップ1202において、出現情報ファイル作成プログラム123を起動し、ワークエリア170に格納されているテキスト150におけるn-gramの出現回数、先頭回数、末尾回数および単独回数を、それぞれ出現情報ファイル151に格納されている該当n-gramの出現回数、先頭回数、末尾回数および単独回数に加算し、ワークエリア170に格納するとともに出現情報ファイル151として磁気ディスク装置105に格納する。

【0096】以上が、出現情報ファイル作成登録プログラム121の処理手順である。

【0097】次に、図11に示したステップ1102で文書登録制御プログラム111により起動される出現確率ファイル作成登録プログラム124の処理手順について、図16のPAD図を用いて説明する。

【0098】出現確率ファイル作成登録プログラム124では、まずステップ1600で出現確率算出プログラム125を起動し、ワークエリア170に格納されている各n-gramの出現情報から、各n-gramの単独確率、先頭確率および末尾確率を算出し、ワークエリア170へ格納する。

次に、ステップ1601において、出現確率ファイル作成プログラム126を起動し、ワークエリア170に格納されている各n-gramの単独確率、先頭確率および末尾確率を出現確率ファイル152として磁気ディスク装置105に格納する。

【0099】以上が、出現確率ファイル作成登録プログラム124の処理手順である。

【0100】次に、図11に示したステップ1103で文書登録制御プログラム111により起動される出現回数ファイル作成登録プログラム127の処理手順について、図25に示すPAD図を用いて説明する。

【0101】出現回数ファイル作成登録プログラム127では、まずステップ2500で出現回数計数プログラム128を起動し、図12のステップ1200でワークエリア170に格納した全ての単一文字種文字列の中から、長さが1から単一文字種文字列自体の長さmまでのn-gramを抽出し、登録対象文書におけるそれらの出現回数を計数し、ワークエリア170に格納する。

【0102】次に、ステップ2501において、出現回数ファイル作成プログラム129を起動し、ステップ2500で計数した各n-gramの出現回数を登録対象文書の

識別番号（以下、文書番号と呼ぶ）とともに出現回数ファイル153として磁気ディスク装置105に格納する。次に、図10に示したステップ1004でシステム制御プログラム110により起動される検索制御プログラム112による類似文書検索の処理手順について、図13のPAD図を用いて説明する。

【0103】検索制御プログラム112では、まずステップ1300で検索条件式解析プログラム130を起動し、キーボード101から入力された検索条件式を解析し、検索条件式のパラメータとして指定された種文書の文書番号を抽出する。

【0104】次に、ステップ1301で類似文書検索プログラム131を起動し、上記検索条件式解析プログラム130により抽出された文書番号の種文書に対し、磁気ディスク装置105に格納されているテキスト150中の各文書の類似度を算出する。

【0105】最後に、ステップ1302において、検索結果出力プログラム132を起動し、上記類似文書検索プログラム131で算出された各文書の類似度に基づいて、検索結果を出力する。

【0106】以上が、検索制御プログラム112による文書検索の処理手順である。

【0107】次に、図13に示したステップ1301で検索制御プログラム112により起動される類似文書検索プログラム131の処理手順について、図14のPAD図を用いて説明する。

【0108】類似文書検索プログラム131では、まずステップ1400で種文書読み込みプログラム140を起動し、検索条件式解析プログラム130によって検索条件式から抽出された文書番号の種文書を磁気ディスク装置105中のテキスト150からワークエリア170に読み込む。

【0109】ここで、種文書は、テキスト150中に格納されている文書を読み込むだけでなく、キーボード101から直接入力することも可能であり、フロッピーディスク104、CD-ROM装置（図1には示していない）や通信回線等を用いて、他の装置から入力するような構成を取ることも可能であり、また、全文検索システム等による検索結果から入力するような構成を取ることも可能であり、さらには、検索結果出力プログラム132の出力から種文書を選択する構成を取ることも可能である。

【0110】次に、ステップ1401において、共有ライブラリ160の単一文字種文字列抽出プログラム161を起動し、上記種文書読み込みプログラム140で読み込んだ種文書のテキストを、文字種境界で分割して単一文字種文字列を取得し、ワークエリア170に格納する。

【0111】そして、ステップ1402において、後述する特徴文字列抽出プログラム141を起動し、上記単

一文字種文字列抽出プログラム161で取得した単一文字種文字列から、特徴文字列を抽出する。

【0112】次に、ステップ1403において、種文書内出現回数計数プログラム145を起動し、上記特徴文字列抽出プログラム141で取得した特徴文字列の、種文書内での出現回数を計数する。

【0113】次に、ステップ1404において、出現回数取得プログラム146を起動し、上記特徴文字列抽出プログラム141で取得した特徴文字列のテキスト150中の各文書における出現回数を取得する。

【0114】最後に、ステップ1405において、類似度算出プログラム148を起動し、上記特徴文字列抽出プログラム141で取得した各特徴文字列に対する、上記種文書内出現回数取得プログラム145で取得した種文書内出現回数と、上記出現回数取得プログラム146で取得したテキスト150中の各文書における出現回数から、種文書とテキスト150内の各文書との類似度を算出する。

【0115】本実施例では、類似度の算出に、前述の類似度算出式(1)を用いるが、他の方法を用いても構わない。この類似度算出式(1)を用いて、前述の文書1「・・・携帯電話の使用時のマナーが問題になる。・・・」が種文書として指定された場合の文書iの類似度 $S(i)$ を算出すると、次のようになる。

【0116】 $S(1) = 1.0$

$S(2) = 0.262$

$S(3) = 0.048$

$S(4) = 0.0$

以上が、類似文書検索プログラム131の処理手順である。

【0117】次に、図14に示したステップ1402において、類似文書検索プログラム131により起動される特徴文字列抽出プログラム141の処理手順について、図17のPAD図を用いて説明する。

【0118】特徴文字列抽出プログラム141では、ステップ1700において、図14に示したステップ1401における単一文字種文字列抽出プログラム161により、ワークエリア170に格納されている全ての単一文字種文字列を取得する。

【0119】次に、ステップ1701において、上記ステップ1700で取得した全ての単一文字種文字列に対して、次のステップ1702～1704を繰り返し実行する。

【0120】すなわち、ステップ1702では、ステップ1700で取得した単一文字種文字列の文字種を判定し、その文字種が漢字やカタカナである場合には、ステップ1703を実行し、漢字やカタカナ以外の場合には、ステップ1704を実行する。

【0121】ステップ1703では、後述する分割確率比較特徴文字列抽出プログラム142を起動し、漢字や

カタカナの単一文字種文字列から特徴文字列を抽出する。

【0122】ステップ1704では、漢字やカタカナ以外の単一文字種文字列そのものを特徴文字列として抽出する。

【0123】そして、最後にステップ1705において、上記ステップ1702やステップ1703で抽出された特徴文字列をワークエリア170へ格納する。

【0124】以上が、特徴文字列抽出プログラム141の処理手順である。

【0125】以下、図14に示した特徴文字列抽出プログラム141の処理手順について具体例を用いて説明する。

【0126】図27に、前述の文書1「・・・携帯電話の使用時のマナーが問題になる。・・・」から特徴文字列を抽出する例を示す。

【0127】まず、文書1から単一文字種文字列「・・・」「の」「携帯電話」「の」「使用時」「の」「マナー」「が」「問題」「になる」「の」「・・・」を抽出する。

【0128】次に、これらの単一文字種文字列の文字種を判定し、漢字文字列「携帯電話」、「使用時」および「問題」とカタカナ文字列「マナー」に対して分割確率比較特徴文字列抽出プログラム142により特徴文字列を抽出し、漢字文字列とカタカナ文字列以外の文字列「の」「の」「が」「になる」「の」からは単一文字種文字列そのものを特徴文字列として抽出する。

【0129】以上が、特徴文字列抽出プログラム141の具体的な処理例である。

【0130】次に、図14に示したステップ1404において類似文書検索プログラム131により起動される出現回数取得プログラム146の処理手順を図26のPAD図を用いて説明する。

【0131】出現回数取得プログラム146では、図14に示したステップ1402においてワークエリア170に格納した特徴文字列を取得する(ステップ2600)。

【0132】そして、ワークエリア170に格納されている全ての特徴文字列に対して、ステップ2602を実行する(ステップ2601)。

【0133】ステップ2602では、出現回数ファイル読み込みプログラム147を起動し、テキスト150内の各文書における特徴文字列の出現回数を取得し、ワークエリア170に格納する。

【0134】以上が、出現回数取得プログラム146の処理手順である。

【0135】次に、図17に示したステップ1703において特徴文字列抽出プログラム141により起動される分割確率比較特徴文字列抽出プログラム142の処理手順について、図18のPAD図を用いて説明する。

【0136】分割確率比較特徴文字列抽出プログラム142は、ステップ1800において、最後に特徴文字列が抽出された末尾の文字位置（以下、最新分割点と呼ぶ）LSの初期値を0に設定する。

【0137】そして、図17に示したステップ1703において、入力された単一文字種文字列の文字列長が予め定められた長さ以上のとき、次のステップ1802～1809までを繰り返し実行する（ステップ1801）。

【0138】ステップ1802では、後述する分割確率算出プログラム143を起動し、単一文字種文字列の先頭から  $i$  文字目の分割確率  $P(i)$  と、 $(i+1)$  文字目の分割確率  $P(i+1)$  を算出する。

【0139】次に、ステップ1803において、上記分割確率算出プログラム143で算出した  $P(i)$  と  $P(i+1)$  の値を比較し、 $P(i)$  が  $P(i+1)$  よりも大きい場合にはステップ1804を実行し、 $P(i)$  が  $P(i+1)$  よりも小さい場合にはステップ1806を実行し、 $P(i)$  と  $P(i+1)$  が等しい場合にはステップ1808を実行する。

【0140】ステップ1804では、単一文字種文字列の先頭から  $i$  文字目までの文字列を特徴文字列として抽出する。そして、ステップ1805において、最新分割点LSを  $i$  に設定し、 $i$  の値を1加算する。

【0141】ステップ1806では、単一文字種文字列の先頭から  $(i+1)$  文字目までの文字列を特徴文字列として抽出する。そして、ステップ1807において、最新分割点LSを  $(i+1)$  に設定し、 $i$  の値を2加算する。

【0142】ステップ1808では、それぞれ単一文字種文字列の先頭から  $i$  文字目までの文字列と  $(i+1)$  文字目までの文字列を特徴文字列として抽出する。そして、ステップ1809において、最新分割点LSを  $(i+1)$  に設定し、 $i$  の値を2加算する。

【0143】以上が、分割確率比較特徴文字列抽出プログラム142の処理手順である。

【0144】以下、図18に示した分割確率比較特徴文字列抽出プログラム142の処理手順について具体例を用いて説明する。

【0145】図8に、前述の文書1「・・・携帯電話の使用時のマナーが問題になる。・・・」から抽出された単一文字種文字列「携帯電話」から特徴文字列を抽出する例を示す。

【0146】まず、「携帯電話」における1文字目での分割確率  $P(1)$  は、「携」の単独確率0.01と「電」の先頭確率0.11の積として0.001が算出され、2文字目での分割確率  $P(2)$  は、「携帯」の単独確率0.10と「電」の先頭確率0.36の積として0.036が算出される。次に、これらの分割確率を比較し、値の大きい方で単一文字種文字列「携帯電話」を分割する。この場合、1文字目の分割確率  $P(1)$  ( $=0.000$ ) よりも2文字目の分割確率  $P(2)$  ( $=0.036$ ) の方が大きいので、単一文字種文字列「携帯電話」は

「携帯」と「電話」に分割される。

【0147】また、図20に、上記文書1から抽出した単一文字種文字列「マナー」から特徴文字列を抽出する例を示す。まず、「マナー」における2文字目での分割確率  $P(2)$  は、「マナ」の単独確率0.00と「ー」の単独確率0.00の積として0.00と算出される。次に、3文字目での分割確率  $P(3)$ 、すなわち「マナー」が単一文字種文字列そのものとして出現する確率は「ナー」の末尾確率0.79と1.0の積として0.79と算出される。これらの値を比較して、値の大きい方で単一文字種文字列に分割される。この場合、「マナー」の2文字目での分割確率  $P(2)$  ( $=0.00$ ) よりも3文字目での分割確率  $P(3)$  ( $=0.79$ ) の方が大きいので、3文字目で分割されることになり、結果的に単一文字種文字列「マナー」は分割されないことになる。

【0148】以上が、分割確率比較特徴文字列抽出プログラム142の具体的な処理手順である。

【0149】次に、図18に示したステップ1801において分割確率比較特徴文字列抽出プログラム142により起動される分割確率算出プログラム143の処理手順について、図19のPAD図を用いて説明する。

【0150】分割確率算出プログラム143は、ステップ1900において、図18に示したステップ1801において指定される分割確率の算出位置  $i$  および最新分割点LSを取得する。

【0151】次に、算出位置  $i$  における分割確率  $P(i)$  を算出するために、ステップ1901～1906を実行し、各出現確率を取得する。

【0152】まず、ステップ1901において、図12に示したステップ1201で抽出された  $n$ -gramの長さ  $n$  と分割確率の算出位置  $i$  を比較し、 $(i - LS)$  が  $n$  以下である場合には、ステップ1902を実行し、 $(i - LS)$  が  $n$  よりも大きい場合には、ステップ1903を実行する。

【0153】ステップ1902では、出現確率ファイル読み込みプログラム144を起動し、最新分割点LSから  $i$  文字目までの文字列の単独確率を取得し、分割確率算出位置  $i$  の前方の文字列の出現確率  $Pre(i)$  とする。

【0154】ステップ1903では、出現確率ファイル読み込みプログラム144を起動し、最新分割点LSから  $i$  文字目までの文字列の後方の  $n$ -gramの末尾確率を取得し、分割確率算出位置  $i$  の前方の文字列の出現確率  $Pre(i)$  とする。

【0155】次に、ステップ1904において、単一文字種文字列の文字列長  $Ln$  と分割確率算出位置  $i$  を比較し、 $Ln$  が  $(i+1)$  よりも大きい場合にはステップ1905を実行し、 $Ln$  が  $(i+1)$  と等しい場合には、ステップ1906を実行する。

【0156】ステップ1905では、出現確率ファイル読み込みプログラム144を起動し、 $(i+1)$  文字目の  $1$ -gram



の先頭確率を取得し、分割確率算出位置  $i$  の後方の文字列の出現確率  $\text{Post}(i)$  とする。

【0157】ステップ1906では、出現確率ファイル読み込みプログラム144を起動し、 $(i+1)$ 文字目の1gramの単独確率を取得し、分割確率算出位置  $i$  の後方の文字列の出現確率  $\text{Post}(i)$  とする。

【0158】次に、算出位置  $(i+1)$  における分割確率  $P(i+1)$  を算出するために、ステップ1907～1913を実行し、各出現確率を取得する。

【0159】まず、ステップ1907において、図12に示したステップ1201で抽出された  $n$ -gramの長さ  $n$  と分割確率の算出位置  $i$  を比較し、 $((i+1) - \text{LS})$  が  $n$  以下である場合には、ステップ1908を実行し、 $((i+1) - \text{LS})$  が  $n$  よりも大きい場合には、ステップ1909を実行する。

【0160】ステップ1908では、出現確率ファイル読み込みプログラム144を起動し、最新分割点LSから  $(i+1)$ 文字目までの文字列の単独確率を取得し、分割確率算出位置  $(i+1)$  の前方の文字列の出現確率  $\text{Pre}(i+1)$  とする。

【0161】ステップ1909では、出現確率ファイル読み込みプログラム144を起動し、最新分割点LSから  $(i+1)$ 文字目までの文字列の後方の  $n$ -gramの末尾確率を取得し、分割確率算出位置  $(i+1)$  の後方の文字列の出現確率  $\text{Pre}(i+1)$  とする。

【0162】次に、ステップ1910において、単一文字種文字列の文字列長  $\text{Ln}$  と分割確率算出位置  $i$  を比較し、 $\text{Ln}$  が  $(i+2)$  よりも大きい場合にはステップ1911を実行し、 $\text{Ln}$  が  $(i+2)$  と等しい場合には、ステップ1912を実行し、 $\text{Ln}$  が  $(i+1)$  と等しい場合には、ステップ1913を実行する。

【0163】ステップ1911では、出現確率ファイル読み込みプログラム144を起動し、 $(i+2)$ 文字目の1gramの先頭確率を取得し、分割確率算出位置  $(i+1)$  の後方の文字列の出現確率  $\text{Post}(i+1)$  とする。

【0164】ステップ1912では、出現確率ファイル読み込みプログラム144を起動し、 $(i+2)$ 文字目の1gramの単独確率を取得し、分割確率算出位置  $(i+1)$  の後方の文字列の出現確率  $\text{Post}(i+1)$  とする。

【0165】ステップ1913では、分割確率算出位置  $(i+1)$  の後方の文字列の出現確率  $\text{Post}(i+1) = 1$  とする。

【0166】次に、ステップ1914において、上記ステップ1901～1903で取得した  $\text{Pre}(i)$  と上記ステップ1904～1906で取得した  $\text{Post}(i)$  の積を算出位置  $i$  における分割確率  $P(i)$  とし、上記ステップ1907～1909で取得した  $\text{Pre}(i+1)$  と上記ステップ1910～1913で取得した  $\text{Post}(i+1)$  の積を算出位置  $(i+1)$  における分割確率  $P(i+1)$  として、それぞれワークエリア170に格納する。

【0167】以上が、分割確率算出プログラム143の

処理手順である。

【0168】以下、図19に示した分割確率算出プログラム143の処理手順について具体例を用いて説明する。

【0169】図28に前述の文書1「・・・携帯電話の使用時のマナーが問題になる。・・・」から抽出された単一文字種文字列「携帯電話」の分割確率を算出する例を示す。なお、本図に示す例では、出現確率ファイル152に格納されている  $n$ -gram長を2とし、分割確率を算出する  $i$ 文字目を1文字目とする。すなわち、1文字目での分割確率  $P(1)$  および2文字目での分割確率  $P(2)$  を算出するものとして、以下の説明を行なう。

【0170】まず、分割確率の算出位置である1文字目までの文字列の単独確率が出現確率ファイル600に格納されているかどうかを確認するために、出現確率ファイル600に格納されている  $n$ -gram長2と分割確率算出位置1を比較する。その結果、格納されている  $n$ -gram長の方が大きいので、1文字目までの文字列「携」の単独確率0.01を出現確率ファイル600より取得する。

【0171】次に、分割確率の算出位置の後方に何文字存在するかを確認するために、単一文字種文字列「携帯電話」の文字列長4と分割確率算出位置1を比較する。その結果、2文字以上の文字列「帯電話」が存在するため、「帯」の先頭確率0.11を出現確率ファイル600から取得する。そして、「携」の単独確率0.01と「帯」の先頭確率0.11の積を算出し、1文字目での分割確率  $P(1) = 0.001$  を得る。

【0172】同様に、分割確率の算出位置である2文字目までの文字列の単独確率が出現確率ファイル600に格納されているかどうかを確認するために、出現確率ファイル600に格納されている  $n$ -gram長2と分割確率算出位置2を比較する。その結果、格納されている  $n$ -gram長と算出位置が等しいので、2文字目までの文字列「携帯」の単独確率0.10を出現確率ファイル600より取得する。

【0173】次に、分割確率の算出位置の後方に何文字存在するかを確認するために、単一文字種文字列「携帯電話」の文字列長4と分割確率算出位置2を比較する。その結果、2文字以上の文字列「電話」が存在するため、「電」の先頭確率0.36を出現確率ファイル600から取得する。そして、「携帯」の単独確率0.10と「電」の先頭確率0.36の積を算出し、2文字目での分割確率  $P(2) = 0.036$  を得る。

【0174】以上が、分割確率算出プログラム143の具体的な処理手順である。

【0175】以上が、本発明の第一の実施例である。

【0176】本実施例では、出現情報ファイル151と出現確率ファイル152に格納する  $n$ -gramの長さとして2を用いて、特徴文字列抽出プログラム143の処理手順を説明したが、この長さとして1や3等の固定値を用いてもよいし、データベース中の出現回数等の情報に基

づき可変長としてもよいし、単一文字種文字列自体の長さ $m$ としてもよいし、さらには、それらの組み合わせであっても、同様に特徴文字列抽出の処理を行なうことができるのは明らかであろう。

【0177】また、本実施例では、種文書の内容に類似する文書を検索するものとして特徴文字列抽出プログラム143の処理手順を説明したが、この種文書の代わりに、文章が指定されたとしても同様に特徴文字列を抽出することができ、類似文書検索を行なうことができるのは明らかであろう。

【0178】また、本実施例では、単一文字種文字列の先頭から $n$ 文字目までの分割確率と $(n+1)$ 文字目までの分割確率を比較することで特徴文字列を抽出する例を用いて、分割確率比較特徴文字列抽出プログラム142の処理手順を説明したが、単一文字種文字列の末尾から、それぞれ $n$ 文字目までの分割確率と $(n+1)$ 文字目までの分割確率を比較しても、さらには、単一文字種文字列中の $m$ 文字( $m$ は1以上の整数)と $n$ 文字の分割確率を比較しても、同様に、文書の特徴を表す特徴文字列の抽出が行えることは明らかであろう。

【0179】なお、本実施例においては、漢字やカタカナの単一文字種文字列に対する分割確率比較特徴文字列抽出プログラム142を含む構成として説明したが、漢字あるいはカタカナを含まないデータベースを対象とする場合等には、対応する分割確率比較特徴文字列抽出プログラム142を含まない構成としてもよいし、漢字やカタカナ以外に対応する分割確率比較特徴文字列抽出プログラム142を含む構成としてもよいし、従来技術2で示したように、各文字種に対応する特徴文字列抽出プログラムを含む構成であってもよい。

【0180】また、本実施例においては、単一文字種文字列から特徴文字列を抽出する構成としたが、特定の文字種間を境界として前後に跨る部分文字列から特徴文字列を抽出することにより、例えば、「F1」や「ビタミンC」、「W杯」、「ケイ素」等の文字列を検索に用いることができ、さらに高精度な類似文書検索を実現することも可能となる。

【0181】また、本実施例における出現情報ファイル作成登録プログラム121では、文字種境界を単語の区切れ目とみなし、各 $n$ -gramの先頭回数、末尾回数および単独回数を計数するものとしたが、付属語、すなわち助詞や助動詞等を単語の区切れ目の候補とみなし、各 $n$ -gramの先頭回数、末尾回数および単独回数を計数してもよい。

【0182】さらに、本実施例においては、出現情報ファイル151を図5に示した表形式で作成されるものとしたが、この方法では、対象とする $n$ -gram長が増大するにともない、 $n$ -gram種類が増加するため、分割確率ファイル作成登録プログラム124の処理に長大な時間を要することになる。この問題は、特徴文字列に対して、検

索用のインデックスを付加することにより解決できる。これにより、 $n$ -gram種類が増加しても、高速に登録処理を実現することができる。この特徴文字列に対する検索用インデックスとしては、全文検索用インデックス153を用いてもよいし、「特開平8-329112号公報」等に開示されているような単語インデックス方式を用いてもよい。この問題は、出現確率ファイル152および出現回数ファイル153においても発生するが、同様に検索用のインデックスを付加することで解決することができる。

【0183】さらに、本実施例においては、文書登録時に出現確率ファイル作成登録プログラム124を起動し、出現確率ファイル152を作成する構成としたが、類似文書検索時の分割確率比較特徴文字列抽出プログラム142実行時に、出現情報ファイル151に格納されている各 $n$ -gramの出現情報から該当する出現確率を算出することにより、磁気ディスク105に格納するファイルを削減することも可能である。

【0184】また、本実施例においては、特徴文字列抽出プログラム141により抽出された特徴文字列を用いた類似文書検索システムについて説明したが、種文書から特徴文字列を抽出する特徴文字列抽出システムとして用いることも可能であるし、「特開平8-153121号公報」に示されるような形態素解析により文書中に含まれる単語を抽出し、これを用いて文書を自動的に分類するシステムに用いることも可能である。

【0185】ただし、第一の実施例における分割確率比較特徴文字列抽出プログラム142は、 $i$ 文字目での分割確率 $P(i)$ と $(i+1)$ 文字目での分割確率 $P(i+1)$ を比較し、その値の大きい方で分割するため、全ての単一文字種文字列から $(i+1)$ 文字以下の特徴文字列が抽出されてしまい、 $(i+1)$ 文字より長い単語が誤って分割されてしまうという問題がある。

【0186】以下、第一の実施例で $(i+1)$ 文字より長い単語が誤って分割されてしまうという問題が生じる例を図22に示す具体例を用いて説明する。なお、本図では、漢字で構成される単一文字種文字列「北海道」を対象とし、分割確率算出位置 $i$ の初期値を1とする。

【0187】分割確率比較特徴文字列抽出プログラム142では、まず、ステップ2200において、前述した分割確率算出プログラム143を起動し、1文字目の分割確率 $P(1)$ と2文字目の分割確率 $P(2)$ を算出する。本図に示した例では、単一文字種文字列「北海道」の1文字目で「北」と「海道」に分割される確率は、1-gram「北」の単独確率0.03と2-gram「海道」の単独確率0.00の積として $P(1)=0.000$ と算出される。同様に、2文字目で「北海」と「道」に分割される確率は、2-gram「北海」の単独確率0.03と1-gram「道」の単独確率0.12の積 $P(2)=0.004$ として算出される。

【0188】次に、ステップ2201において、上記ステップ2200で算出された $P(1)$ と $P(2)$ のうち、値の大

きい方を分割点とし、単一文字種文字列の先頭から分割点までの文字列を特徴文字列として抽出する。本図に示した例では、 $P(2)$ の方が $P(1)$ よりも大きいので、2文字目で単一文字種文字列「北海道」を分割し、2文字目までの文字列「北海」を特徴文字列として抽出する。

【0189】次に、ステップ2202において、最後に特徴文字列が抽出された末尾の文字位置（以下、最新分割点と呼ぶ） $LS$ を2に設定し、最新分割点以降の単一文字種文字列「道」を対象に特徴文字列抽出処理を継続する。

【0190】次に、ステップ2203において、単一文字種文字列「道」の文字列長 $l$ は、予め定められた長さ2未満であるため、文字列「道」が特徴文字列として抽出される。この結果、「・・・道の駅と呼ばれるサービスエリアが国道沿いに建設されることになった。・・・」等という文書が類似文書として誤って検索されてしまうことになる。

【0191】以上が、第一の実施例における分割確率比較特徴文字列抽出プログラム142の処理例である。本図に示した例では、1文字目と2文字目の分割確率 $P(1)$ と $P(2)$ を比較し、値の大きい方を分割点とするため、単一文字種文字列「北海道」から「北海」と「道」が特徴文字列として抽出されてしまい、種文書の中心概念からずれた文書が類似文書として検索されてしまう。

【0192】このために、本発明を適用した類似文書検索システムの第二の実施例では、単一文字種文字列から特徴文字列を抽出する際に算出された分割確率が所定値（以下、分割閾値と呼ぶ）よりも高い場合にのみ、比較処理を行なうことにより、 $(i+1)$ 文字より長い特徴文字列を抽出できるようにする。

【0193】本実施例は、第一の実施例（図1）とほぼ同様の構成を取るが、分割確率比較特徴文字列抽出プログラム142の処理手順が異なり、図21のPAD図に示すように、ステップ2100～2104が追加される。

【0194】以下、第二の実施例における分割確率比較特徴文字列抽出プログラム142aの処理手順について図21のPAD図を用いて説明する。

【0195】分割確率比較特徴文字列抽出プログラム142aでは、ステップ1800において、最新分割点 $LS$ の初期値を0に設定する。

【0196】そして、特徴文字列の抽出対象となる単一文字種文字列の文字列長が予め定められた長さ以上のとき、次のステップ1802～1807、ステップ2101～2103までを繰り返し実行する（ステップ2100）。

【0197】ステップ1802では、分割確率算出プログラム143を起動し、単一文字種文字列の先頭から $i$ 文字目の分割確率 $P(i)$ と、 $(i+1)$ 文字目の分割確率 $P(i+1)$ を算出する。

【0198】次に、ステップ2100において、上記分割確率算出プログラム143で算出された分割確率 $P(i)$ 、 $P(i+1)$ の値および予め定められた分割閾値 $Th$ の値を比較し、最大のものを抽出する。この結果、分割確率 $P(i)$ が抽出されたならばステップ1804を実行し、分割確率 $P(i+1)$ が抽出された場合にはステップ1806を実行し、分割閾値 $Th$ が抽出された場合にはステップ2101を実行する。

【0199】ステップ1804では、単一文字種文字列の先頭から $i$ 文字目までの文字列を特徴文字列として抽出する。そして、ステップ1805において、最新分割点 $LS$ を $i$ に設定し、 $i$ の値を1加算する。

【0200】ステップ1806では、単一文字種文字列の先頭から $(i+1)$ 文字目までの文字列を特徴文字列として抽出する。そして、ステップ1807において、最新分割点 $LS$ を $(i+1)$ に設定し、 $i$ の値を2加算する。

【0201】ステップ2101では、分割確率の算出位置 $i$ と単一文字種文字列の文字列長 $ln$ とを比較し、 $(i+1)$ が文字列長 $ln$ よりも小さい場合には、ステップ2102を実行し、 $(i+1)$ が文字列長 $ln$ 以上であるならば、ステップ2103を実行する。

【0202】ステップ2102では、分割確率の算出位置 $i$ の値を1加算する。

【0203】ステップ2103では、単一文字種文字列そのものを特徴文字列として抽出する。そして、ステップ2104において、最新分割点 $LS$ を文字列長 $ln$ に設定し、 $i$ の値を1加算する。

【0204】以上が、分割確率比較特徴文字列抽出プログラム142aの処理手順である。

【0205】以下、第二の実施例における分割確率比較特徴文字列抽出プログラム142aの処理手順をそれぞれ図23に示す具体例で説明する。なお、本図では、漢字で構成される単一文字種文字列「北海道」を対象とし、分割閾値 $Th$ を0.050とし、分割確率算出位置 $i$ の初期値を1として分割確率比較特徴文字列抽出プログラム142aの処理手順を説明する。

【0206】分割確率比較特徴文字列抽出プログラム142aでは、まず、ステップ2200において、前述した分割確率算出プログラム143を起動し、1文字目の分割確率 $P(1)$ と2文字目の分割確率 $P(2)$ を算出し、 $P(1)=0.000$ および $P(2)=0.004$ を得る。

【0207】次にステップ2301において、上記ステップ2200で算出した分割確率 $P(1)$ 、 $P(2)$ および分割閾値 $Th$ うち、最大のものを抽出する。この結果、分割閾値 $Th$ が最大であるので、ステップ2302において、分割確率の算出位置 $i(i=1)$ と単一文字種文字列「北海道」の文字列長 $ln(=3)$ を比較する。この結果、分割確率の算出位置 $i$ の方が小さいので、 $i$ の値を1加算する。

【0208】そして、ステップ2304において、2文字目での分割確率 $P(2)$ と3文字目での分割確率 $P(3)$ を算

出する。この例では、2文字目で「北海」と「道」に分割される確率は、2-gram「北海」の単独確率0.03と1-gram「道」の単独確率0.12の積 $P(2)=0.004$ として算出され、3文字目までの「北海道」として出現する確率は、2-gram「北海」の先頭確率と2-gram「海道」の末尾確率の積 $P(3)=0.465$ として算出される。

【0209】次に、ステップ2305において、上記ステップ2304で算出した分割確率 $P(2)$ 、 $P(3)$ および分割閾値 $Th$ のうち、最大のものを抽出する。この結果、 $P(3)$ が最大であるので、3文字目「北海道」までが特徴文字列として抽出される。

【0210】以上説明したように、本実施例によれば、分割確率が分割閾値よりも高い場合にのみ、比較処理を行なうようにすることにより、本来分割されることのない位置での分割を削減することができる。このため、第一の実施例で抽出されていた不適切な特徴文字列を大幅に削減することが可能となる。そのため、種文書が表わす概念を適切に検索できるとともに、高速に類似文書を検索することができるようになる。

【0211】次に、本発明の第三の実施例について図29を用いて説明する。

【0212】第一の実施例および第二の実施例においては、特徴文字列として抽出される可能性のある全ての文字列を出現回数ファイル153中に格納しておく必要があるため、文字列の種類の増加に伴い、データベース内の各文書における出現回数の取得に長大な時間を要するとともに、必要な磁気ディスク容量が増加してしまう。

【0213】本発明を適用した類似文書検索システムの第三の実施例は、種文書から抽出した特徴文字列に対するデータベース内の各文書における出現回数の取得に、出現回数ファイル153を用いずに、全文検索用インデクスを利用することにより上記必要な磁気ディスク容量を低減する方式である。

【0214】すなわち、本実施例によれば、第一の実施例におけるデータベース内の各文書における出現回数の取得に全文検索システムを利用することにより、文字列の種類数が多いデータベースに対しても高速な類似文書検索を実現することが可能となる。さらに、出現回数ファイル153を全文検索用インデクスで代用するため、本類似文書検索システムを全文検索システムと組み合わせて実現した場合に、第一の実施例に比べ必要となる磁気ディスク容量を削減できることになる。

【0215】本実施例は、第一の実施例（図1）とほぼ同様の構成を取るが、類似文書検索プログラム131中の出現回数取得プログラム146を構成する出現回数ファイル読み込みプログラム147が異なる。このプログラムの代わりに、図29に示すように全文検索プログラム2902が用いられる。

【0216】以下、本実施例における処理手順のうち、第一の実施例とは異なる出現回数取得プログラム146

aの処理手順について、図30を用いて説明する。

【0217】ここで、第一の実施例における出現回数取得プログラム146（図26）と異なる点は、出現回数取得ステップ3000だけである。他の処理ステップの処理手順は、第一の実施例で説明した通りである。

【0218】出現回数取得ステップ3000では、特徴文字列抽出プログラム141によりワークエリア170に格納された特徴文字列を全文検索プログラム2902で検索することにより、テキスト150内の各文書における該特徴文字列の出現回数を取得する。

【0219】本実施例の出現回数取得ステップ3000で用いる全文検索プログラム2902としては、どのような方式を適用しても構わない。例えば、「特開昭64-35627号公報」（以下、従来技術4と呼ぶ）で開示されているようなn-gramインデクス方式を用いることも可能である。

【0220】この従来技術4によるn-gramインデクス方式では、図29に示すように、文書の登録時に、データベースへ登録する文書のテキストデータからn-gramとそのn-gramのテキスト中における出現位置を抽出し、全文検索用インデクス2901として磁気ディスク装置2900に格納しておく。検索時には指定された検索ターム中に出現するn-gramを抽出し、これらに対応するインデクスを上記磁気ディスク装置2900中の全文検索用インデクス2901から読み込み、インデクス中のn-gramの出現位置を比較し、検索タームから抽出したn-gramの位置関係とインデクス中のn-gramの位置関係が等しいかどうかを判定することによって、指定された検索タームが出現する文書を高速に検索する。

【0221】この方法を用いて、特徴文字列を検索タームとして全文検索プログラム2902へ入力し、該特徴文字列の出現文書とその位置情報を取得することにより、該特徴文字列の各文書における出現回数を取得することが可能となる。

【0222】以下、この従来技術4を用いた出現回数の取得方法を図7と図15を用いて具体的に説明する。なお、本図では、n-gramのnの値を1としている。

【0223】まず、文書の登録時の処理手順を図7を用いて具体的に説明する。データベースに登録するテキスト701がn-gramインデクス作成登録ステップ702に読み込まれ、n-gramインデクス700が作成される。このn-gramインデクス700には、テキスト701に出現する全ての1-gramとテキスト701における1-gramの出現位置が格納される。

【0224】本図に示すテキスト701では、「携」という1-gramはテキスト701内の文書番号2の26文字目に現れるので、n-gramインデクス700には1-gram「携」とこれに対応したかたちで、出現位置（2，26）が格納される。ここで、例えば、（2，26）は、文書番号2の26文字目に出現するということを示して

いる。

【0225】次に、検索時の処理手順を図15を用いて具体的に説明する。本図では、前述の文書1「携帯電話の使用のナーが問題になる。・・・」から抽出された特徴文字列「電話」の出現回数を、前述したn-gramインデクス700から取得する例について示す。

【0226】まず、検索対象となる特徴文字列がn-gram抽出部1500に入力され、特徴文字列中に出現する全てのn-gramとそのn-gramの特徴文字列における出現位置が抽出される。次に、抽出されたn-gramとこれに対応するn-gramの特徴文字列における出現位置がインデクス検索部1501に入力される。インデクス検索部1501では、特徴文字列から抽出されたn-gramに対応するインデクスがn-gramインデクス700から読み込まれ、これらのインデクスの中から文書番号が一致し、かつ特徴文字列中の位置関係と同じ位置関係を持つものが抽出され、検索結果として出力される。

【0227】特徴文字列として「電話」が入力された本図の場合、まず、n-gram抽出部1500において、(1-gram「電」、1-gram位置「1」と(1-gram「話」、1-gram位置「2」)が抽出される。ここで、n-gram位置「1」は検索タームの先頭、n-gram位置「2」はその次の文字位置を示す。

【0228】次に、インデクス検索部1501において、n-gramインデクス700から1-gram「電」と「話」に対応するインデクスが読み込まれる。これらのインデクスにおける出現文書番号が等しく、かつ出現位置がn-gram位置「1」とn-gram位置「2」のように連続するものが、すなわち隣接するものが抽出され検索結果として出力される。

【0229】本図では、1-gram「電」の(2, 28)と1-gram「話」の出現位置(2, 29)が文書番号が同じで、位置が「28」と「29」で隣接するため、n-gram「電話」が文字列として存在することが分かり、文書2中に検索ターム「電話」が出現することが検出される。しかし、1-gram「電」の(3, 11)と1-gram「話」の(3, 15)は隣接していないため、この位置には特徴文字列「電話」が出現しないことが分かる。

【0230】そして、上記インデクス検索部1501から検索結果として出力される出現位置を計数することにより、該当特徴文字列の出現回数を得る。

【0231】以上説明したように、本実施例によれば、出現回数ファイルの特徴文字列検索用インデクスと出現回数ファイルの代わりに、全文検索用インデクスを利用することにより、余分なファイルを増やさずに、高速に類似文書検索を実現することが可能となる。

【0232】次に、本発明の第四の実施例について図31を用いて説明する。

【0233】第一、第二および第三の実施例においては、種文書から抽出された単一文字種文字列の先頭から

n文字目での分割確率と(n+1)文字目での分割確率を比較することで特徴文字列を抽出するものとしたが、出現情報ファイル151と出現確率ファイル152を保持する必要があるため、文字列の種類の増加に伴い、必要な磁気ディスク容量が増加してしまう。

【0234】本発明を適用した類似文書検索システムの第四の実施例は、出現情報ファイル151と出現確率ファイル152を用いずに、出現回数ファイル153を利用することで、上記必要な磁気ディスク容量を低減する方式である。

【0235】本発明を適用した第四の実施例は、第一の実施例(図1)とほぼ同様の構成をとるが、類似文書検索プログラム131を構成する特徴文字列抽出プログラム141が異なり、n-gram抽出プログラム3100と前述の出現回数取得プログラム146で構成される。

【0236】以下、本実施例における処理手順のうち、第一の実施例とは異なる特徴文字列抽出プログラム141aの処理手順について、図32を用いて説明する。

【0237】特徴文字列抽出プログラム141aは、まずステップ3200において、前述の単一文字種文字列抽出プログラム161により、ワークエリア170に格納されている全ての単一文字種文字列を取得する。

【0238】次に、ステップ3201において、上記ステップ3200で取得した全ての単一文字種文字列に対して、次のステップ3202～3205を繰り返し実行する。

【0239】すなわち、ステップ3202では、n-gram抽出プログラム3100を起動し、ステップ3200で取得した単一文字種文字列から、予め定められた長さn(nは1以上の整数)のn-gramを先頭から1文字ずつずらしながら、全てのn-gramを抽出する。

【0240】そして、ステップ3203において、上記n-gram抽出プログラム3100により抽出された全てのn-gramに対して、次のステップ3204を繰り返し実行する。すなわち、ステップ3204では、出現回数取得プログラム146を起動し、上記n-gram抽出プログラム3100により抽出されたn-gramの出現回数を取得する。

【0241】そして、ステップ3205において、上記ステップ3204で取得した各n-gramの出現回数の降順にソートし、上位から予め定められた個数のn-gramを特徴文字列として抽出する。

【0242】以上が、特徴文字列抽出プログラム141aの処理手順である。

【0243】以下、図32に示した特徴文字列抽出プログラム141aの処理手順について具体例を用いて説明する。

【0244】図33に、前述の文書1「・・・。携帯電話の使用時のナーが問題になる。・・・」から特徴文字列を抽出する例を示す。本図に示す例ではn-gramのn

の値を2とし、各単一文字種文字列から2個の2-gramを特徴n-gramとして抽出するものとする。

【0245】まず、文書1から単一文字種文字列「・・・」「。」「」」「携帯電話」「の」「使用時」「の」「マナー」「が」「問題」「になる」「。」「・・・」を抽出する。

【0246】次に、これらの単一文字種文字列の先頭から1文字ずつずらしながら全ての2-gramを抽出し、各2-gramの出現回数の降順にソートする。例えば、単一文字種文字列「携帯電話」からは“携帯”、“帯電”、“電話”の3つの2-gramを抽出し、それぞれデータベース内の出現回数を取得する。この結果、(電話, 5,283)、(携帯, 462)、(帯電, 269)が得られる。ここで(電話, 5,282)は、2-gram“電話”のデータベース内における出現回数が5,283回であることを表わす。

【0247】次に、各単一文字種文字列において、上位2個の2-gramを特徴n-gramとして抽出する。この結果、単一文字種文字列「携帯電話」では(電話, 5,283)、(携帯, 462)が上位2個であるため、“電話”および“携帯”が特徴文字列として抽出される。

【0248】以上が、特徴文字列抽出プログラム141aの具体的な処理例である。

【0249】以上説明したように、本実施例によれば、出現情報ファイル151と出現確率ファイル152を用いずに、出現回数ファイル153を利用することにより、データベース中での実際の出現状況を正確に反映した特徴文字列を抽出することが可能となる。

【0250】なお、本実施例では、単一文字種文字列の先頭から1文字ずつずらしながら予め定められた長さnのn-gramを全て抽出するものとして、n-gram抽出プログラム3100の処理手順を説明したが、単一文字種文字列中の任意のn-gramを抽出してもよいし、さらには、単一文字種文字列中のm-gram(mは1以上の整数)とn-gramを抽出してもよい。さらに、抽出するn-gramの長さnを予め定められたものとしたが、単一文字種文字列の長さにより抽出するnの値を変更してもよいし、単一文字種文字列の文字種により変更してもよい。また、本発明のn-gram抽出手法は、文書の特徴を表すn-gramを抽出することができるため、n-gramを用いた文書の特徴を表すベクトルの算出やn-gramを用いた文書の分類にも適用可能である。

【0251】

【発明の効果】本発明によれば、誤分割が少なくなるように特徴文字列を抽出することができるようになる。これにより、単語辞書を参照せずに類似文書検索を行なった場合でも、意味のまとまった文字列を用いて検索を行なうことができるため、中心概念のずれを低減した類似文書検索を実現できる。

【図面の簡単な説明】

【図1】本発明による類似文書検索システムの第一の実

施例の全体構成を示す図である。

【図2】従来技術3における出現情報ファイルの例を示す図である。

【図3】従来技術3における出現確率ファイルの例を示す図である。

【図4】従来技術3における特徴文字列抽出方法の例を示す図である。

【図5】本発明による出現情報ファイルの例を示す図である。

【図6】本発明による出現確率ファイルの例を示す図である。

【図7】本発明の第三の実施例におけるn-gramインデックスの例を示す図である。

【図8】本発明の第一の実施例における分割確率比較特徴文字列抽出プログラム142を漢字文字列に適用した場合の処理例を示す図である。

【図9】本発明による特徴文字列の抽出方法の例を示す図である。

【図10】本発明の第一の実施例におけるシステム制御プログラム110の処理手順を示すPAD図である。

【図11】本発明の第一の実施例における文書登録制御プログラム111の処理手順を示すPAD図である。

【図12】本発明の第一の実施例における出現情報ファイル作成登録プログラム121の処理手順を示すPAD図である。

【図13】本発明の第一の実施例における検索制御プログラム112の処理手順を示すPAD図である。

【図14】本発明の第一の実施例における類似文書検索プログラム131の処理手順を示すPAD図である。

【図15】本発明の第三の実施例における出現回数取得の例を示す図である。

【図16】本発明の第一の実施例における出現確率ファイル作成登録プログラム124の処理手順を示すPAD図である。

【図17】本発明の第一の実施例における特徴文字列抽出プログラム141の処理手順を示すPAD図である。

【図18】本発明の第一の実施例における分割確率比較特徴文字列抽出プログラム142の処理手順を示すPAD図である。

【図19】本発明の第一の実施例における分割確率算出プログラム143の処理手順を示すPAD図である。

【図20】本発明の第一の実施例における分割確率比較特徴文字列抽出プログラム142をカタカナ文字列に適用した場合の処理例を示す図である。

【図21】本発明の第二の実施例における分割確率比較特徴文字列抽出プログラム142aの処理手順を示すPAD図である。

【図22】本発明の第一の実施例における分割確率比較特徴文字列抽出プログラム142の処理例を示す図である。



【図23】本発明の第二の実施例における分割確率比較特徴文字列抽出プログラム142aの処理例を示す図である。

【図24】本発明による出現回数ファイル作成処理の手順を示す図である。

【図25】本発明の第一の実施例における出現回数ファイル作成登録プログラム127の処理手順を示すPAD図である。

【図26】本発明の第一の実施例における出現回数取得プログラム146の処理手順を示すPAD図である。

【図27】本発明の第一の実施例における特徴文字列抽出プログラム141の処理例を示す図である。

【図28】本発明の第一の実施例における分割確率算出の処理例を示す図である。

【図29】本発明の第三の実施例における類似文書検索プログラム131の構成を示す図である。

【図30】本発明の第三の実施例における出現回数取得プログラム146aの処理手順を示す図である。

【図31】本発明の第四の実施例における特徴文字列抽出プログラム141aの構成を示す図である。

【図32】本発明の第四の実施例における特徴文字列抽出プログラム141aの処理手順を示すPAD図である。

【図33】本発明の第四の実施例における特徴文字列抽出プログラム141aの処理例を示す図である。

#### 【符号の説明】

100…ディスプレイ、  
101…キーボード、  
102…中央演算処理装置（CPU）、  
103…フロッピディスクドライブ（FDD）、  
104…フロッピディスク、  
105…磁気ディスク装置、  
106…主メモリ、

107…バス、  
110…システム制御プログラム、  
111…文書登録制御プログラム、  
112…検索制御プログラム、  
120…テキスト登録プログラム、  
121…出現情報ファイル作成登録プログラム、  
122…出現情報計数プログラム、  
123…出現情報ファイル作成プログラム、  
124…出現確率ファイル作成登録プログラム、  
125…出現確率算出プログラム、  
126…出現確率ファイル作成プログラム、  
127…出現回数ファイル作成登録プログラム、  
128…出現回数計数プログラム、  
129…出現回数ファイル作成プログラム、  
130…検索条件式解析プログラム、  
131…類似文書検索プログラム、  
132…検索結果出力プログラム、  
140…種文書読み込みプログラム、  
141…特徴文字列抽出プログラム、  
142…分割確率比較特徴文字列抽出プログラム、  
143…分割確率算出プログラム、  
144…出現確率ファイル読み込みプログラム、  
145…種文書内出現回数計数プログラム、  
146…出現回数取得プログラム、  
147…出現回数ファイル読み込みプログラム、  
148…類似度算出プログラム、  
150…テキスト、  
151…出現情報ファイル、  
152…出現確率ファイル、  
153…出現回数ファイル、  
160…共有ライブラリ、  
161…同一文字種文字列抽出プログラム、  
170…ワークエリア

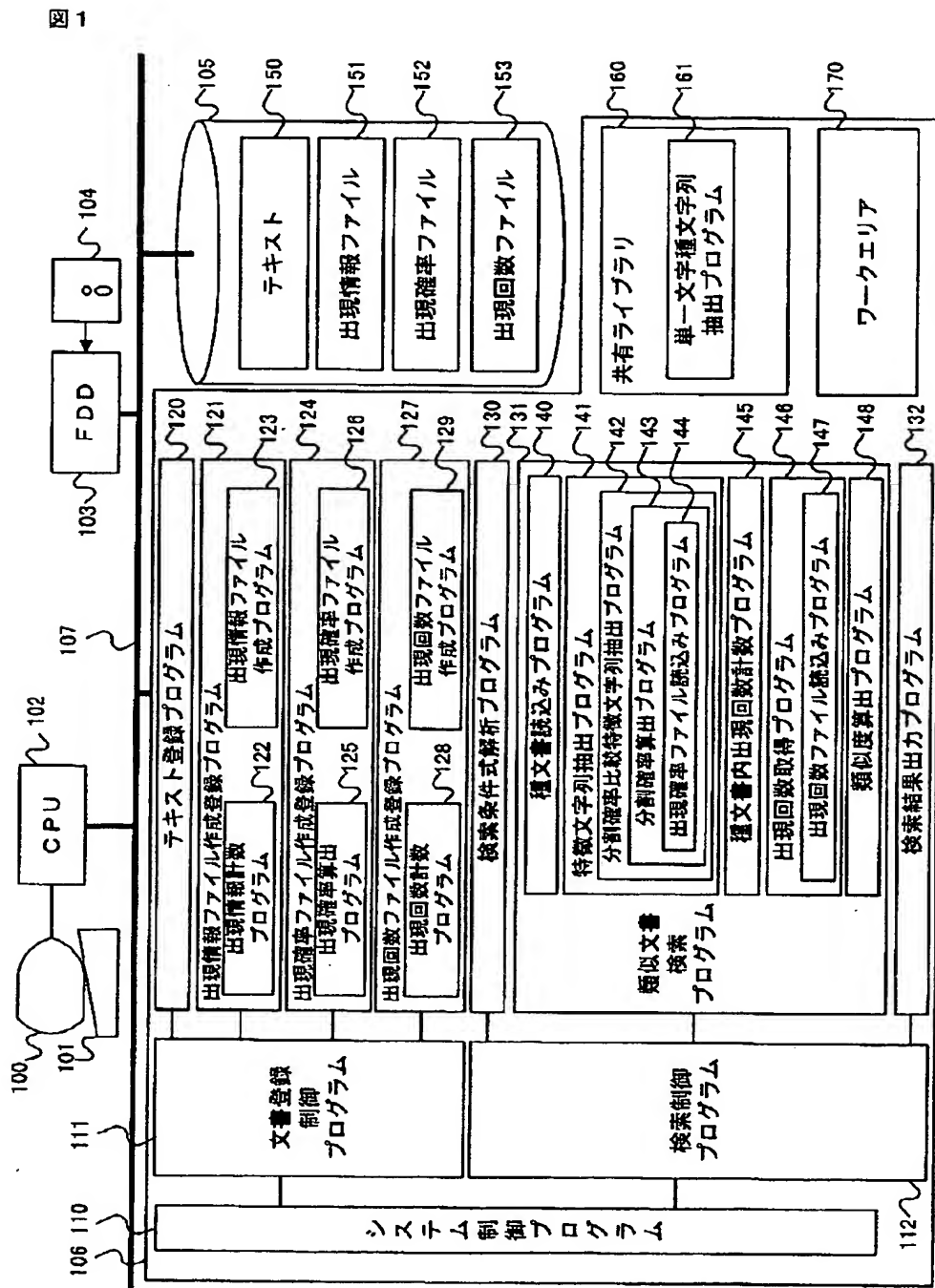
【図2】

図2

No.	1-gram	出現回数	先頭回数	末尾回数
1	一	62,318	0	13,480
2	ナ	28,090	2,653	2,079
3	マ	43,300	15,235	6,179
4	携	4,740	768	492
5	帯	4,703	530	687
6	題	36,338	733	32,342
7	電	38,317	13,794	3,218
8	問	46,216	19,205	11,884
9	用	59,987	5,132	33,600
10	話	18,416	1,105	6,353

200

【図 1】



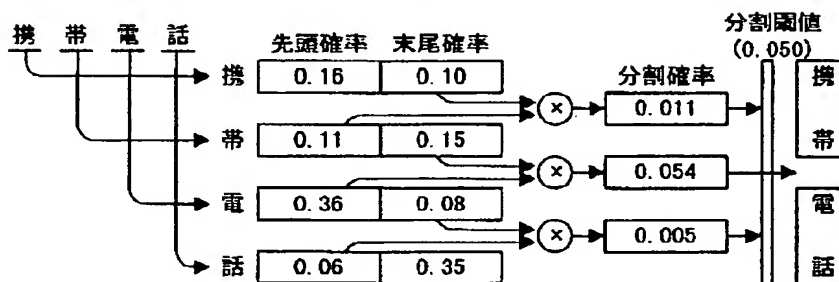
【図3】

図3

No.	1-gram	先頭確率	末尾確率
1	一	0.00	0.22
2	ナ	0.09	0.07
3	マ	0.35	0.14
4	携	0.16	0.10
5	帯	0.11	0.15
6	題	0.02	0.89
7	電	0.36	0.08
8	問	0.42	0.26
9	用	0.09	0.56
10	話	0.06	0.34

【図4】

図4



【図5】

図5

No.	n-gram	出現回数	先頭回数	末尾回数	単独回数
1	一	62,318	0	13,480	0
2	ナ	28,090	2,653	2,079	0
3	マ	43,300	15,235	6,179	0
4	携	4,740	768	492	42
5	帯	4,703	530	687	26
6	題	36,338	733	32,342	332
7	電	38,317	13,794	3,218	218
8	問	46,216	19,205	11,884	884
9	用	59,987	5,132	33,600	768
10	話	18,416	1,105	6,353	211
11	ナ一	3,867	65	3,040	0
12	マナ	122	99	4	0
13	携帯	462	419	52	48
14	使用	2,704	2,156	1,517	517
15	帯電	269	14	4	4
16	電話	5,283	2,538	3,053	1,298
17	問題	29,095	15,280	25,547	13,157

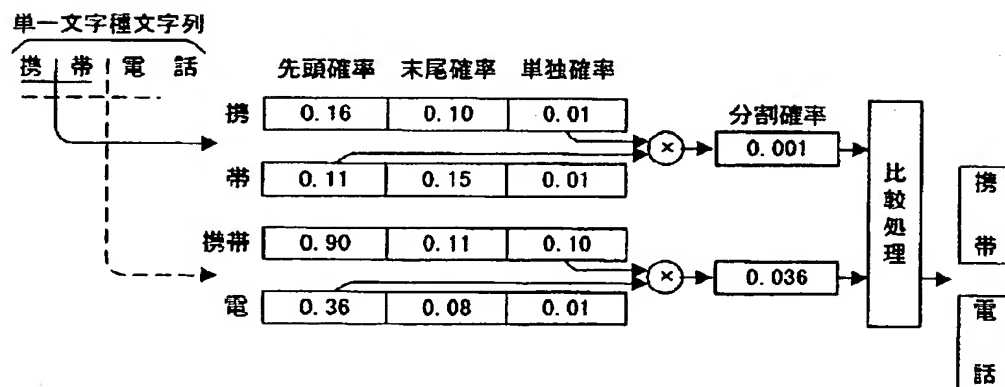
【図6】

図6

No.	n-gram	先頭確率	末尾確率	単独確率
1	一	0.00	0.22	0.00
2	ナ	0.09	0.07	0.00
3	マ	0.35	0.14	0.00
4	携	0.16	0.10	0.01
5	使	0.69	0.70	0.12
6	帯	0.11	0.15	0.01
7	題	0.02	0.89	0.01
8	電	0.36	0.08	0.01
9	問	0.42	0.26	0.02
10	用	0.09	0.56	0.01
11	話	0.06	0.34	0.01
12	ナー	0.02	0.79	0.00
13	マナ	0.81	0.03	0.00
14	携帯	0.90	0.11	0.10
15	使用	0.80	0.56	0.19
16	帯電	0.05	0.01	0.01
17	電話	0.48	0.58	0.25
18	問題	0.53	0.88	0.45

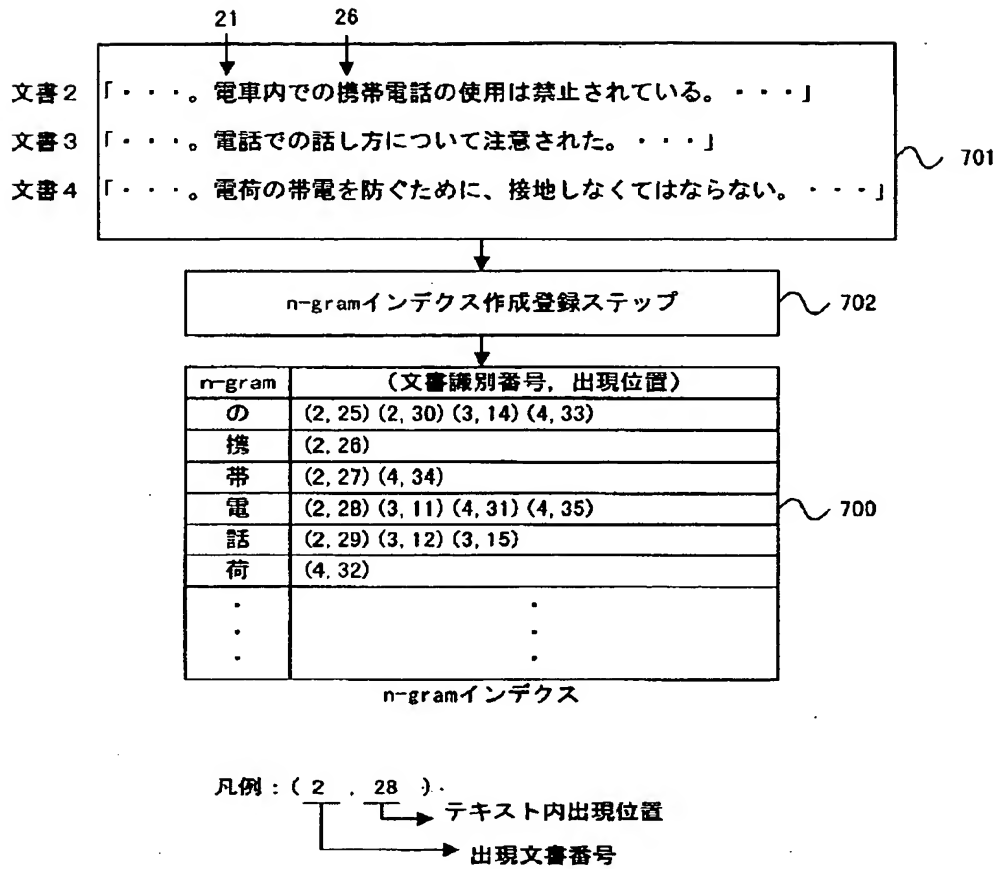
【図8】

図8



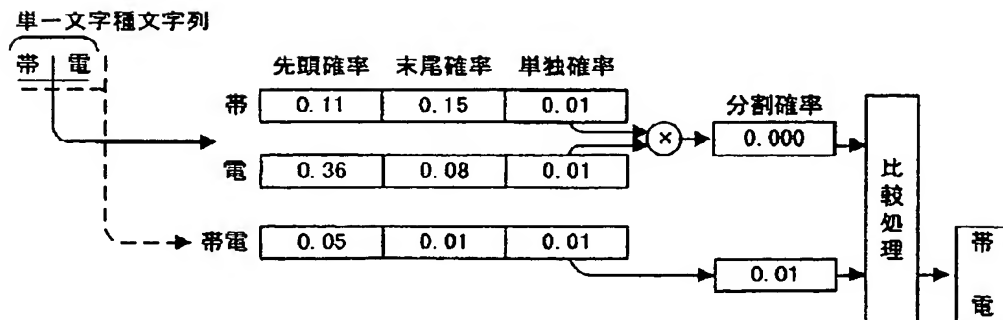
【図7】

図7



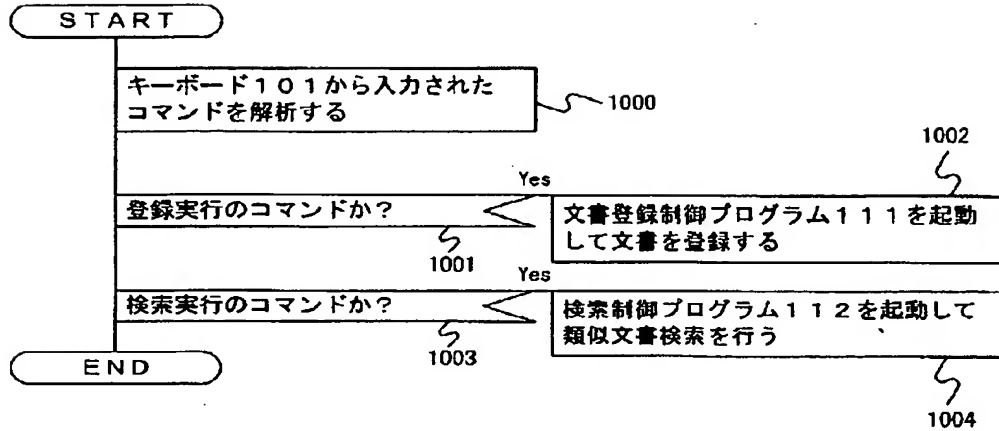
【図9】

図9



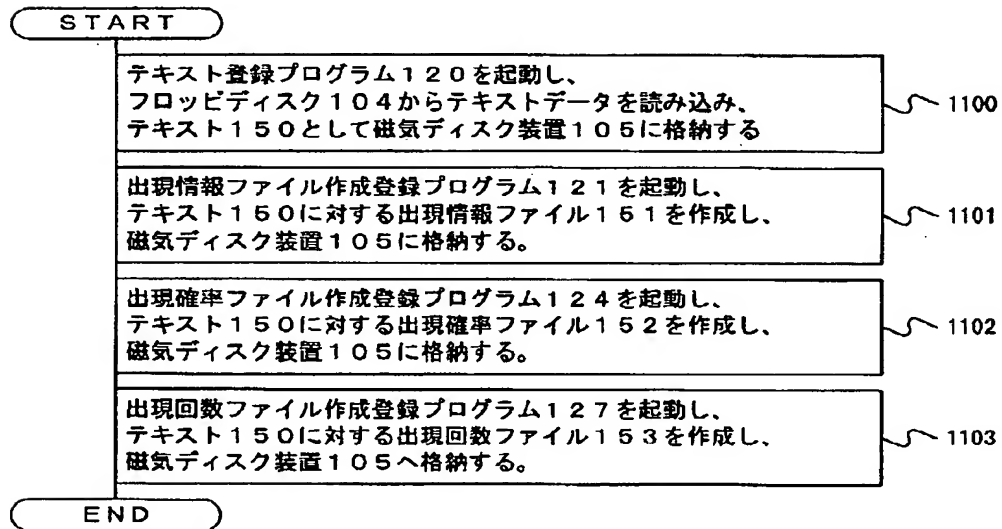
【図10】

図10



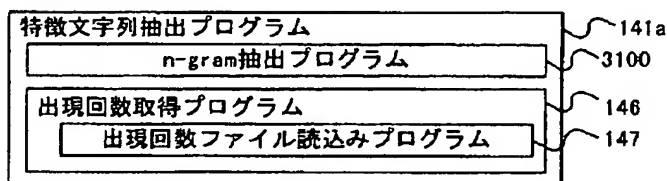
【図11】

図11



【図31】

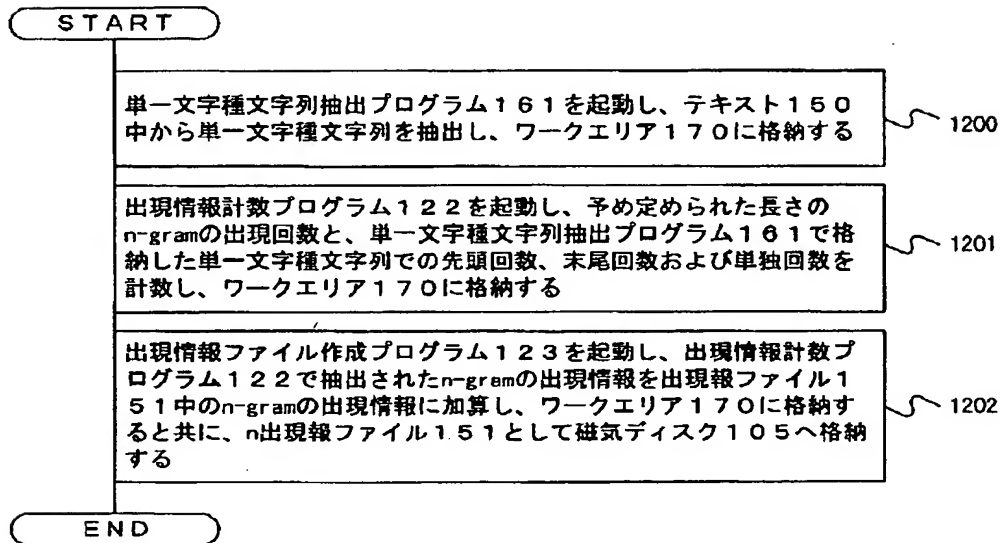
図31





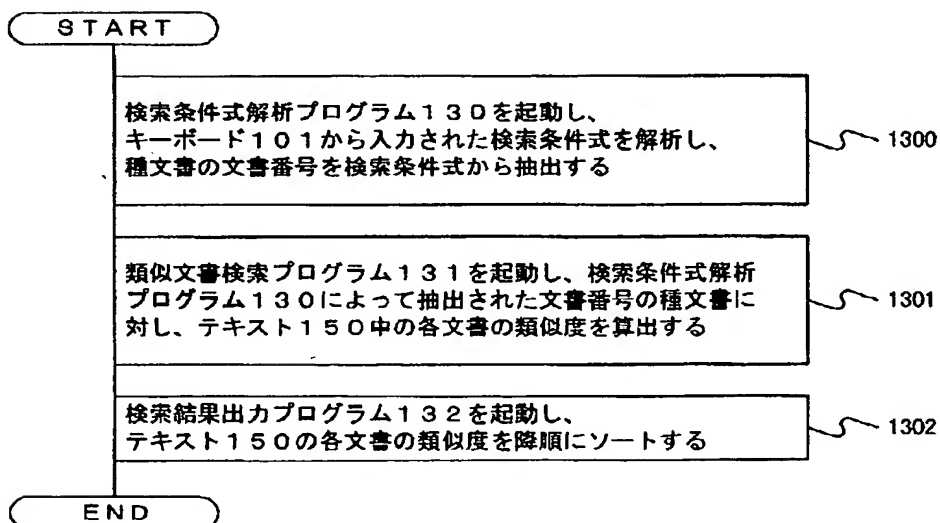
【図12】

図12



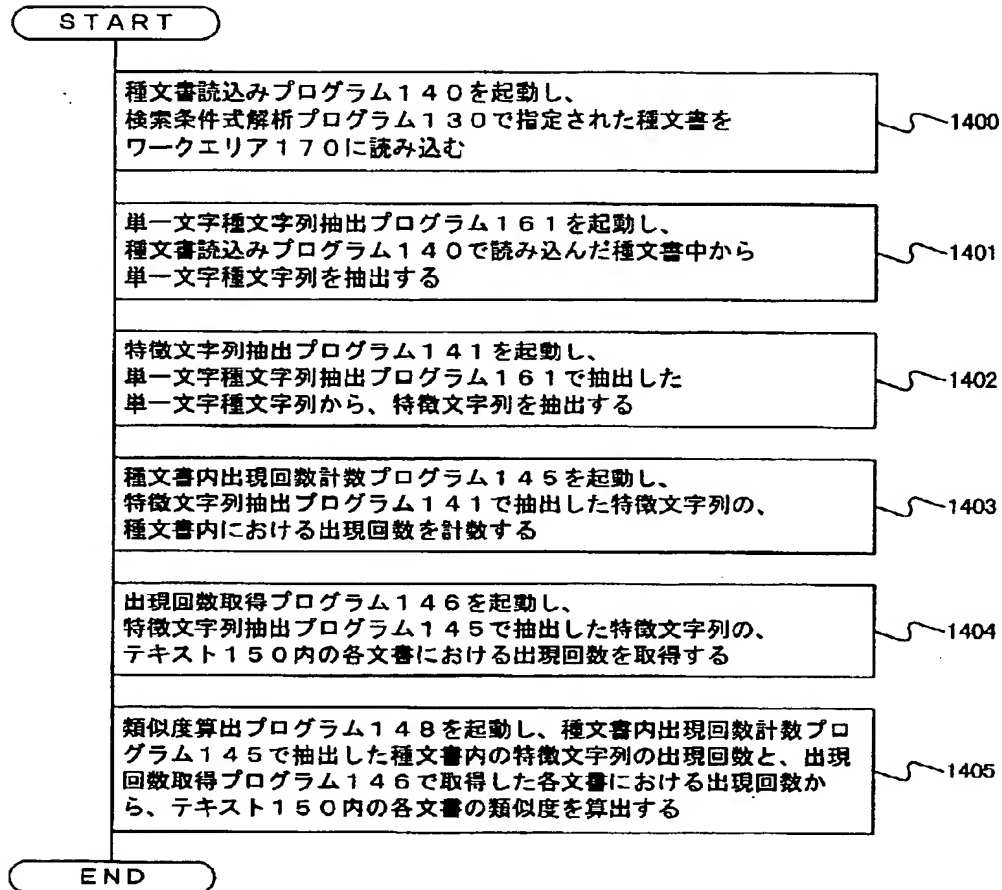
【図13】

図13



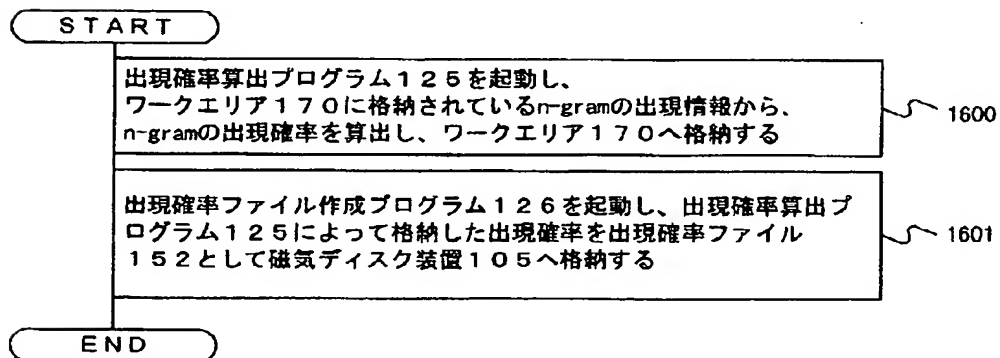
【図14】

図14



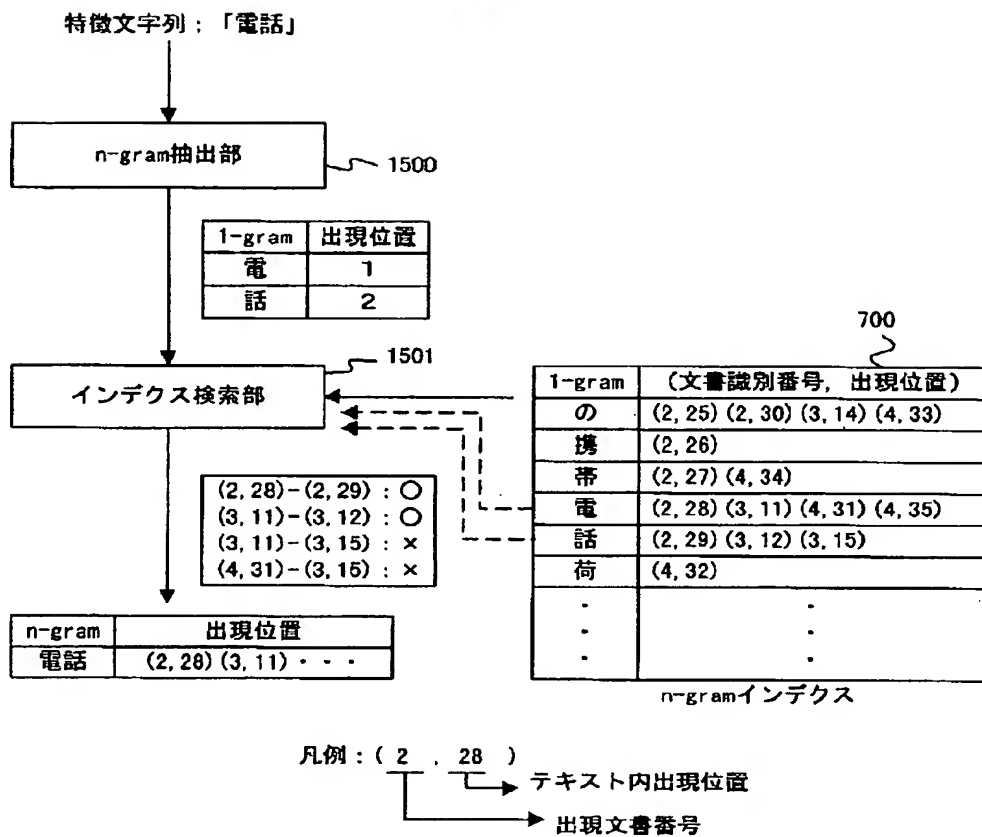
【図16】

図16



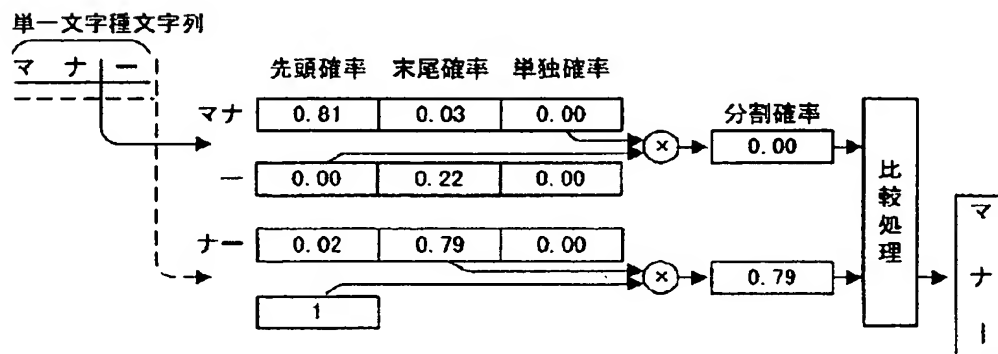
【図15】

図15



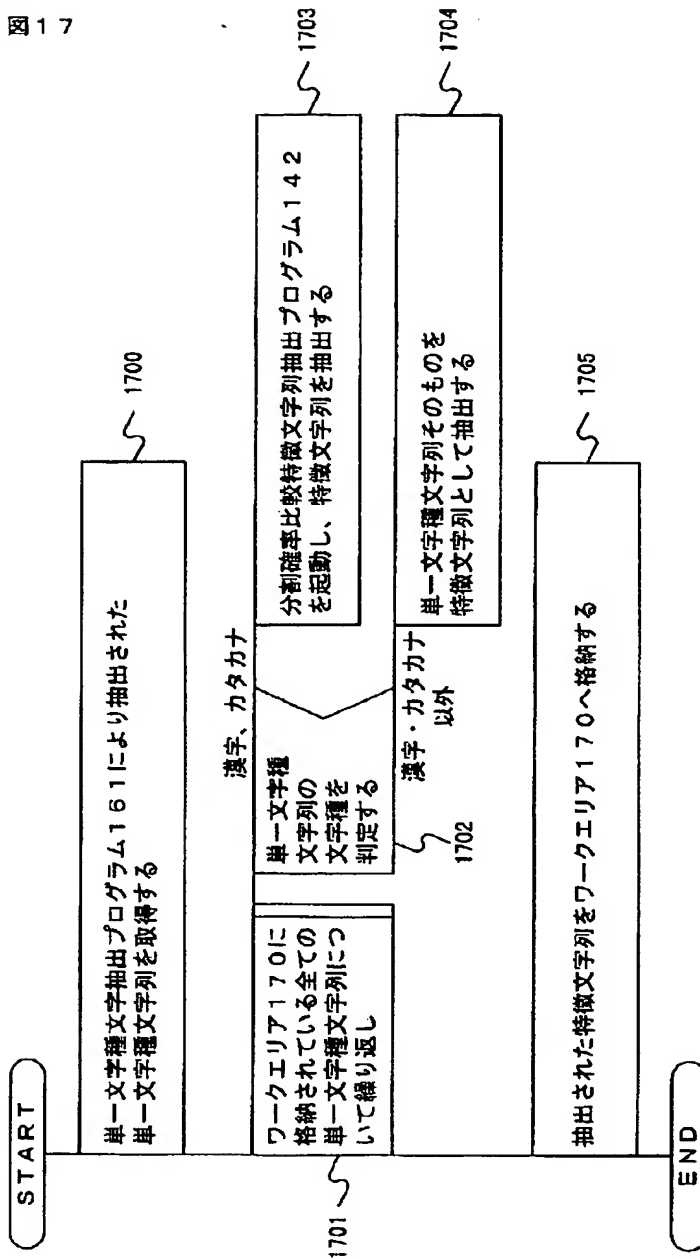
【図20】

図20



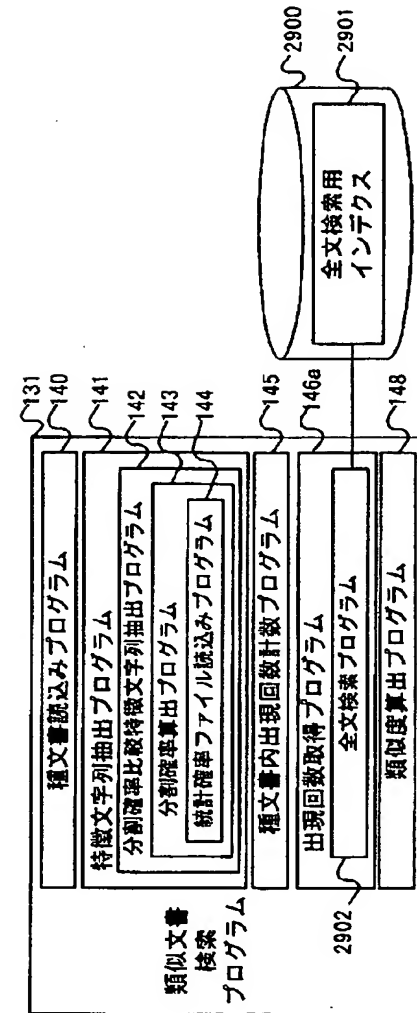
【図17】

図17

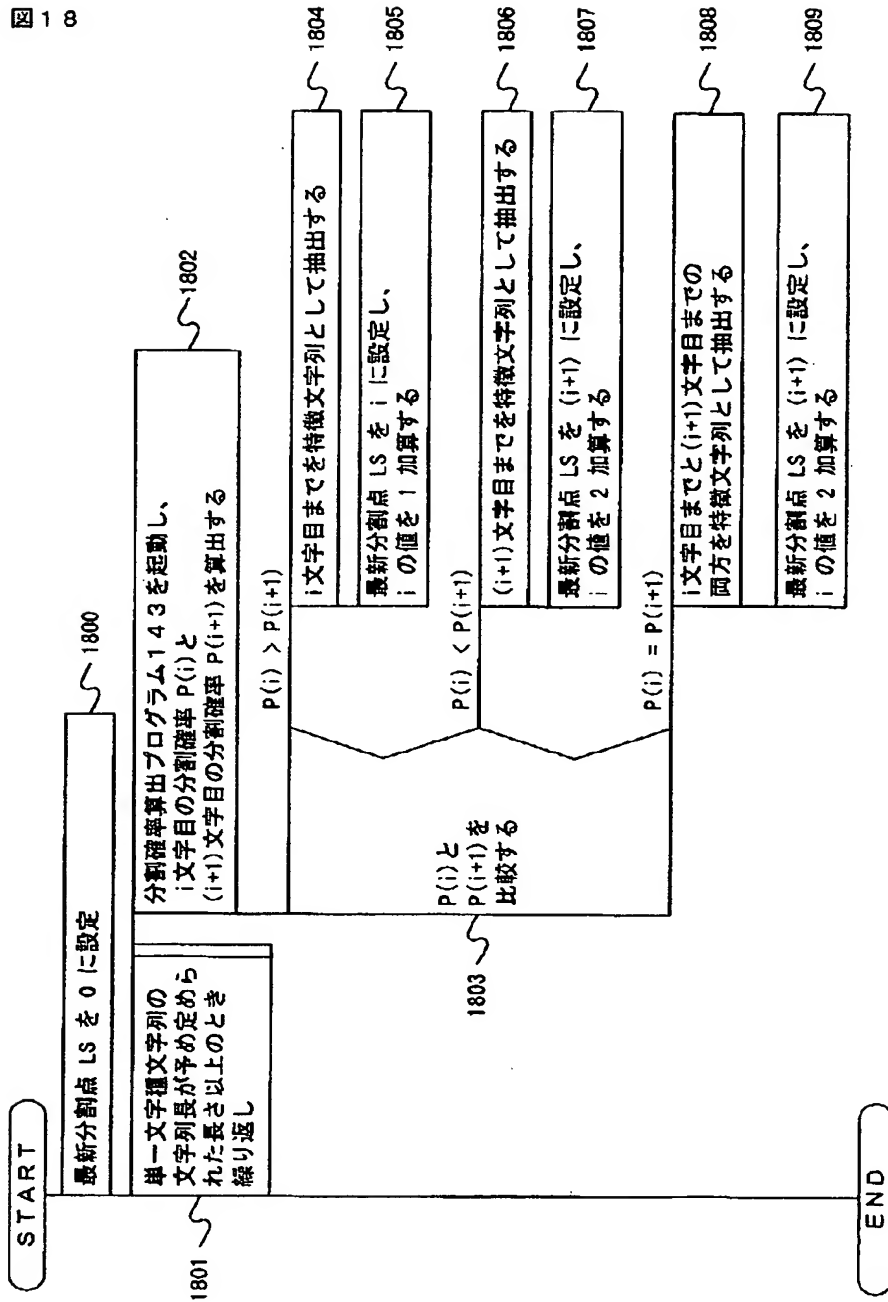


【図29】

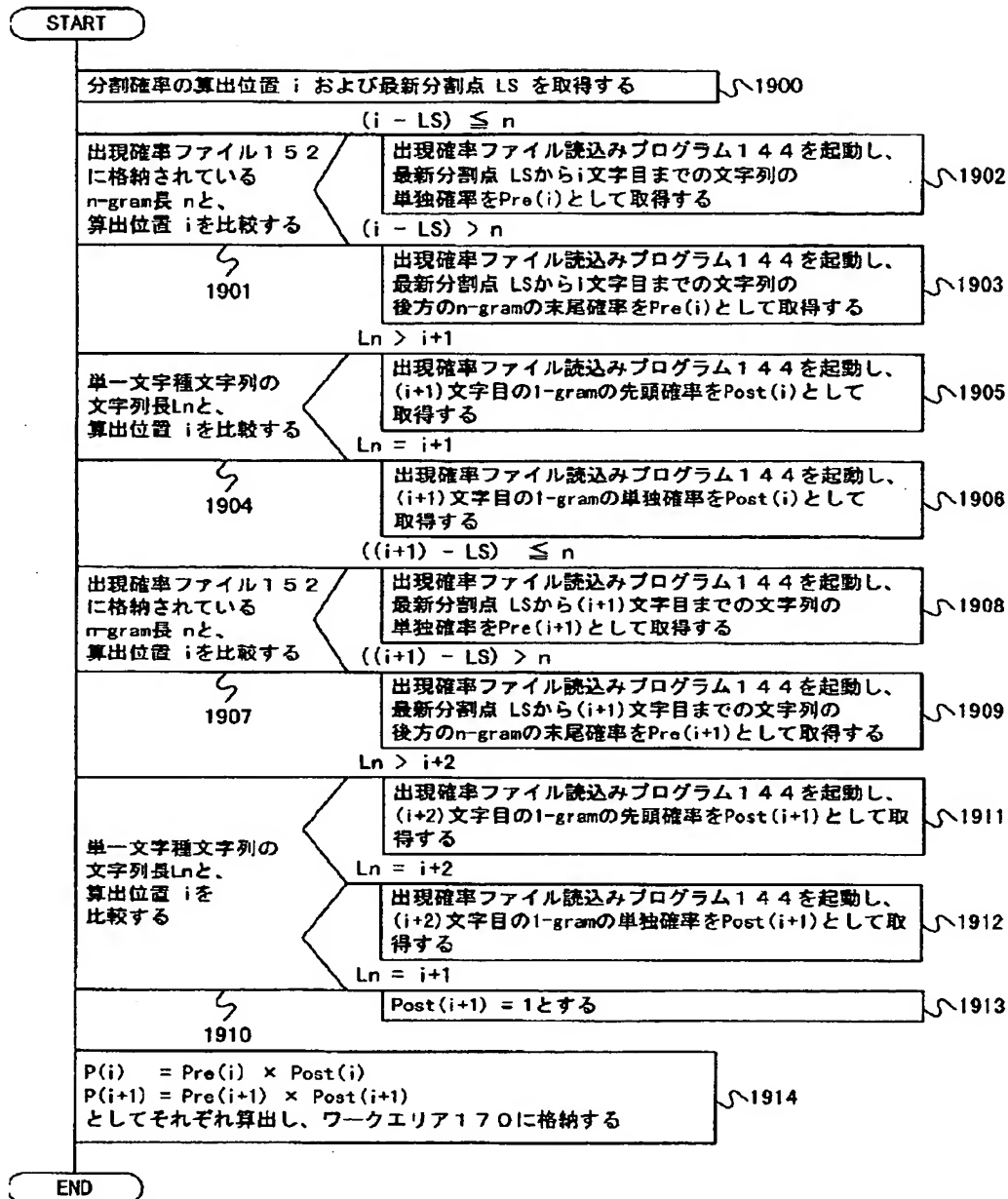
図29



【図18】



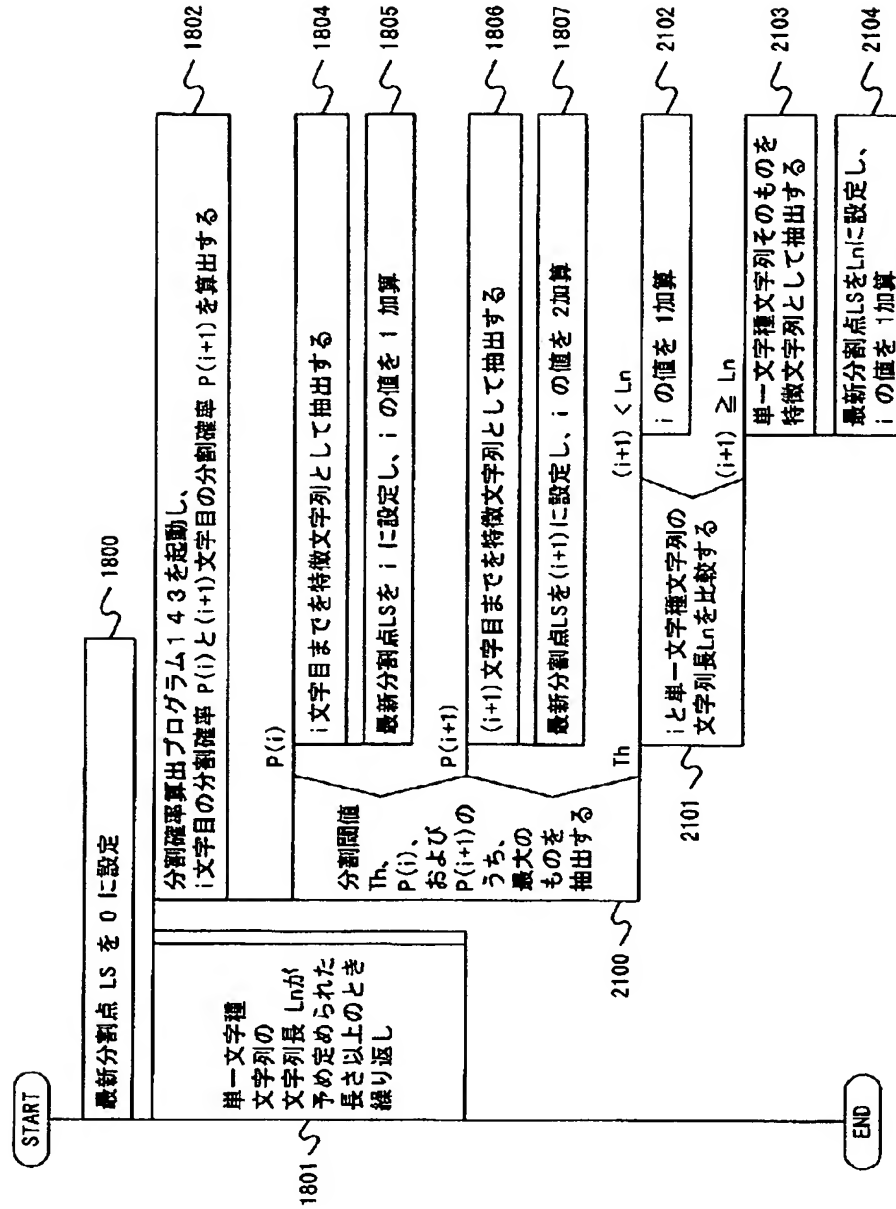
19





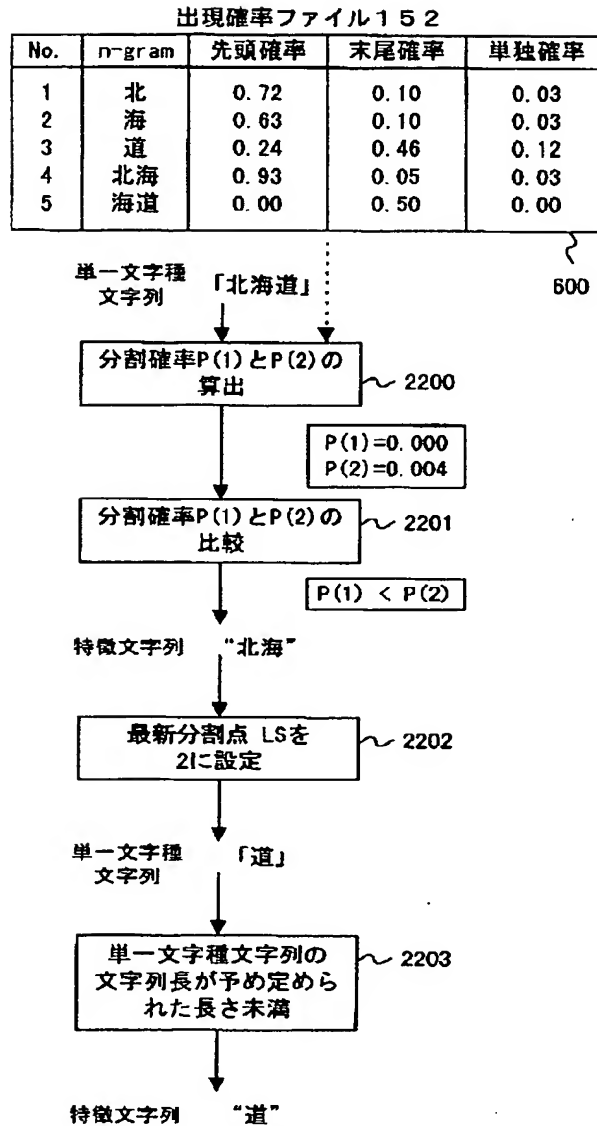
【図21】

図 21



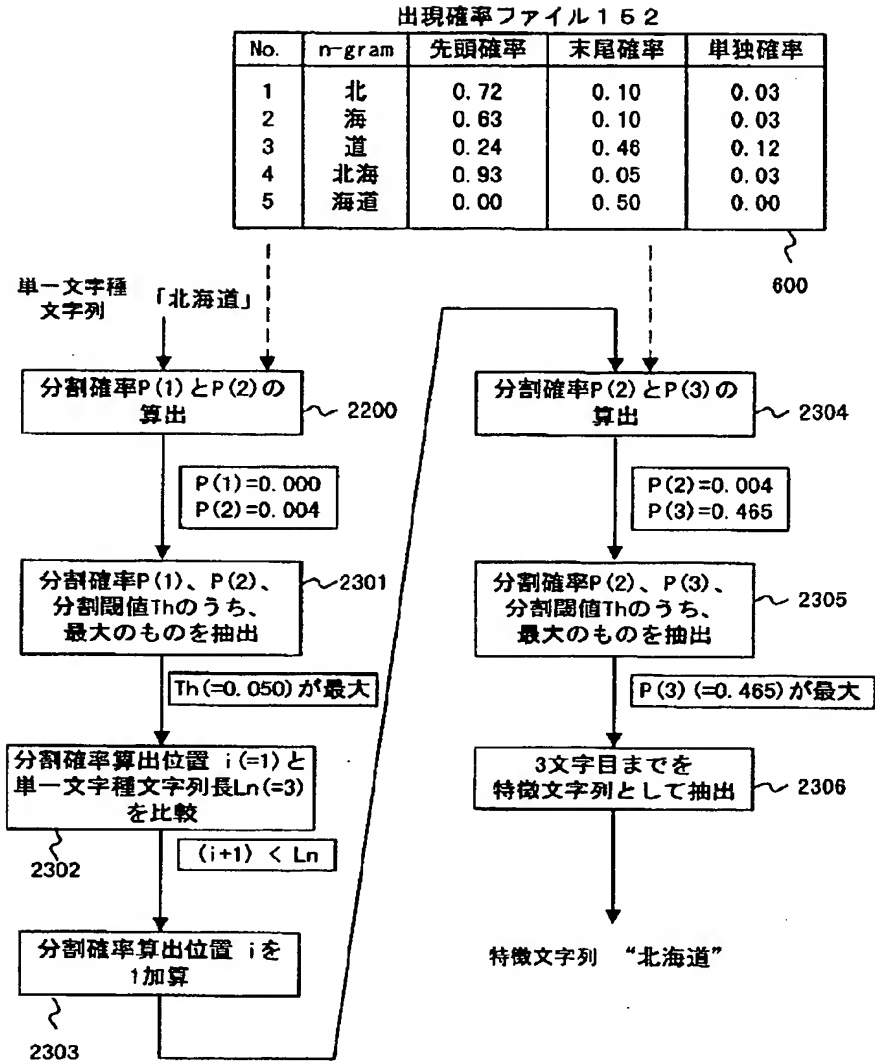
【図22】

図22

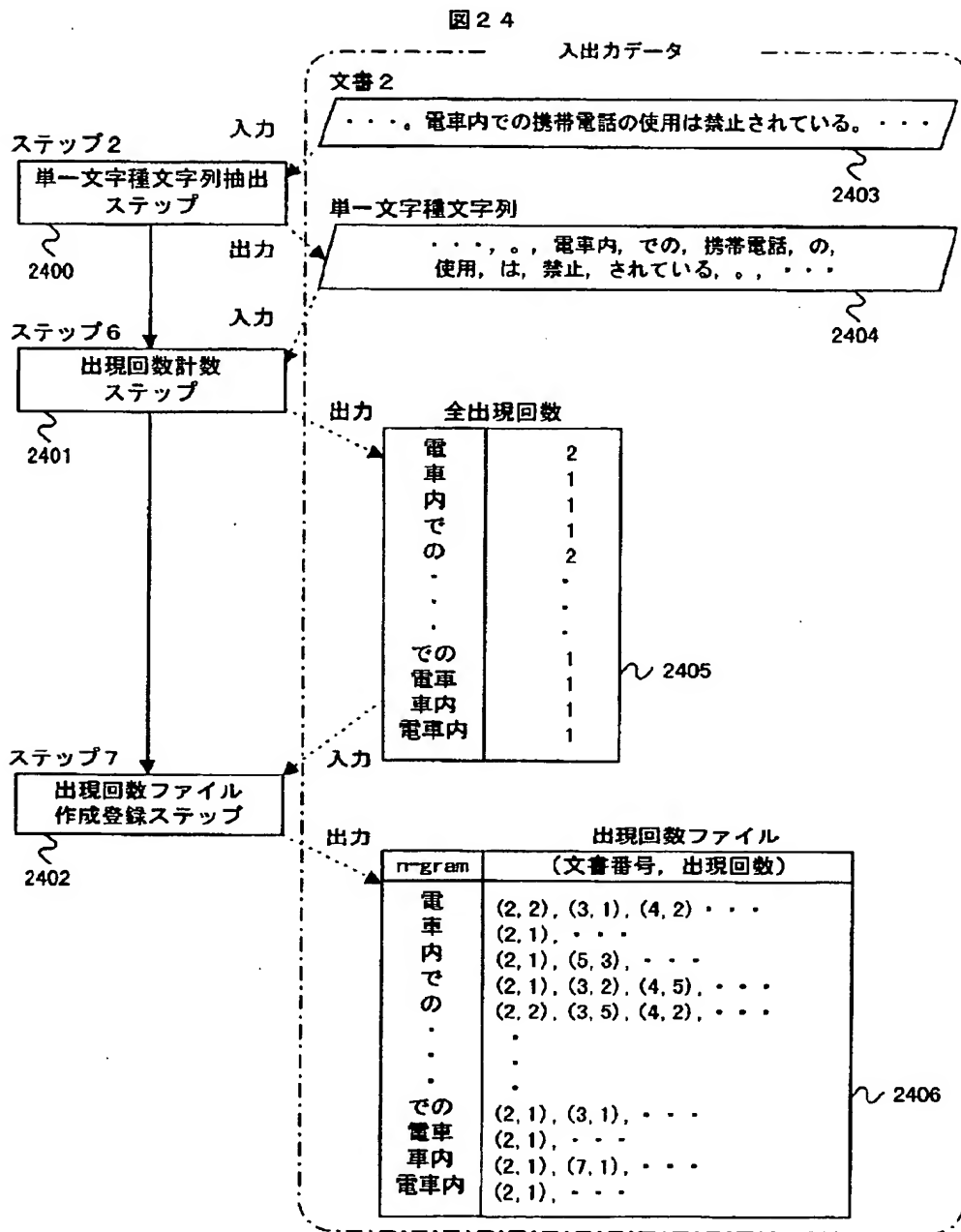


【図23】

図23

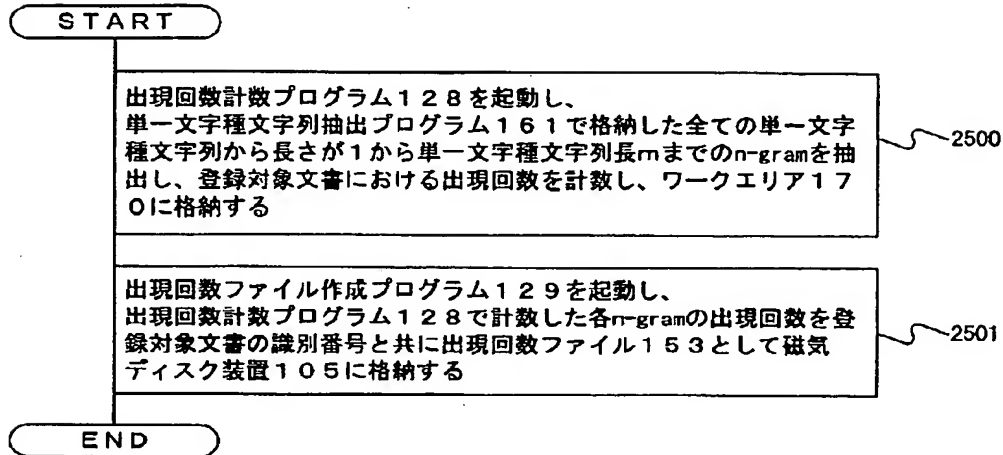


【図24】



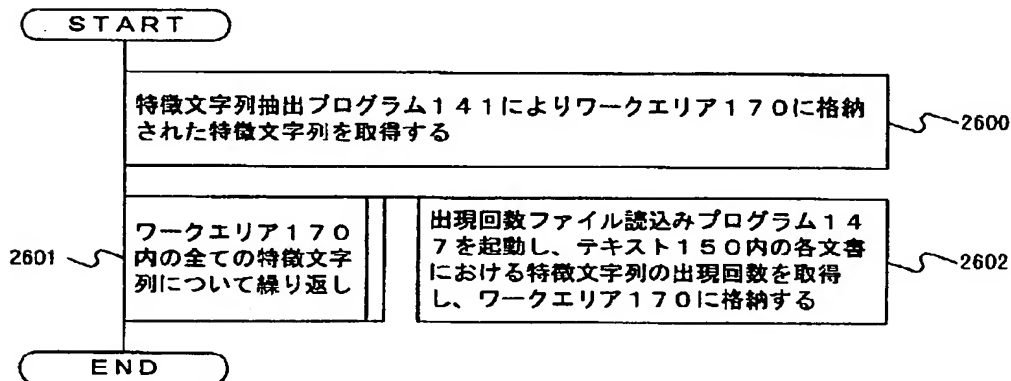
【図25】

図25



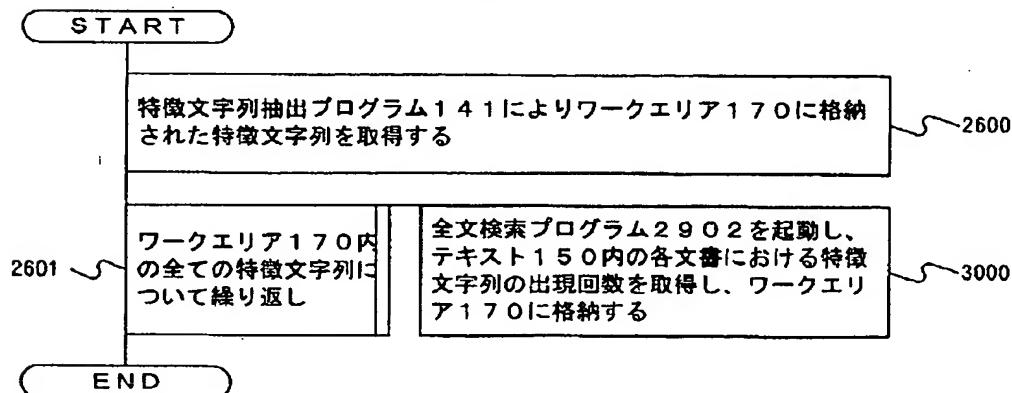
【図26】

図26



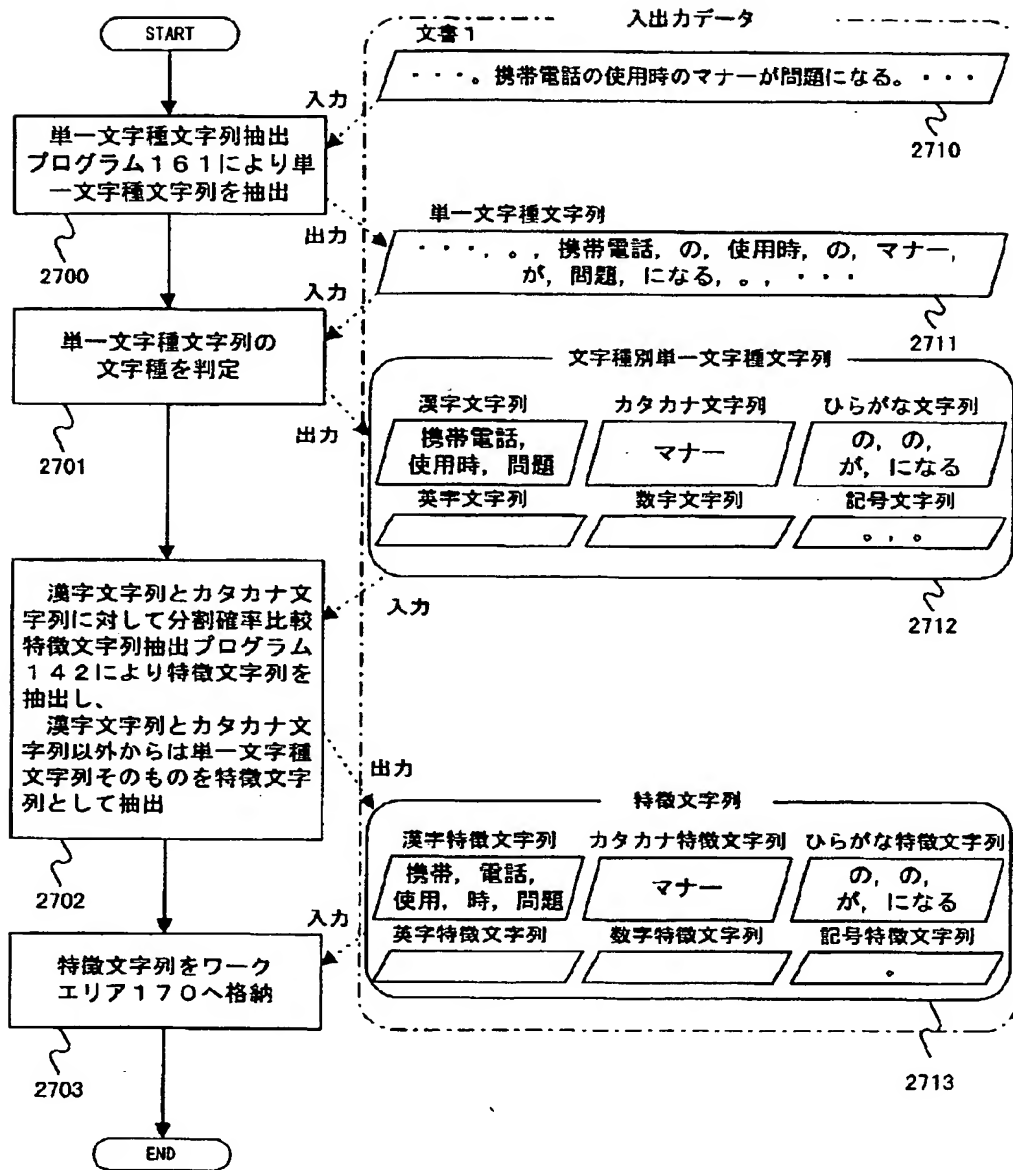
【図30】

図30



【図27】

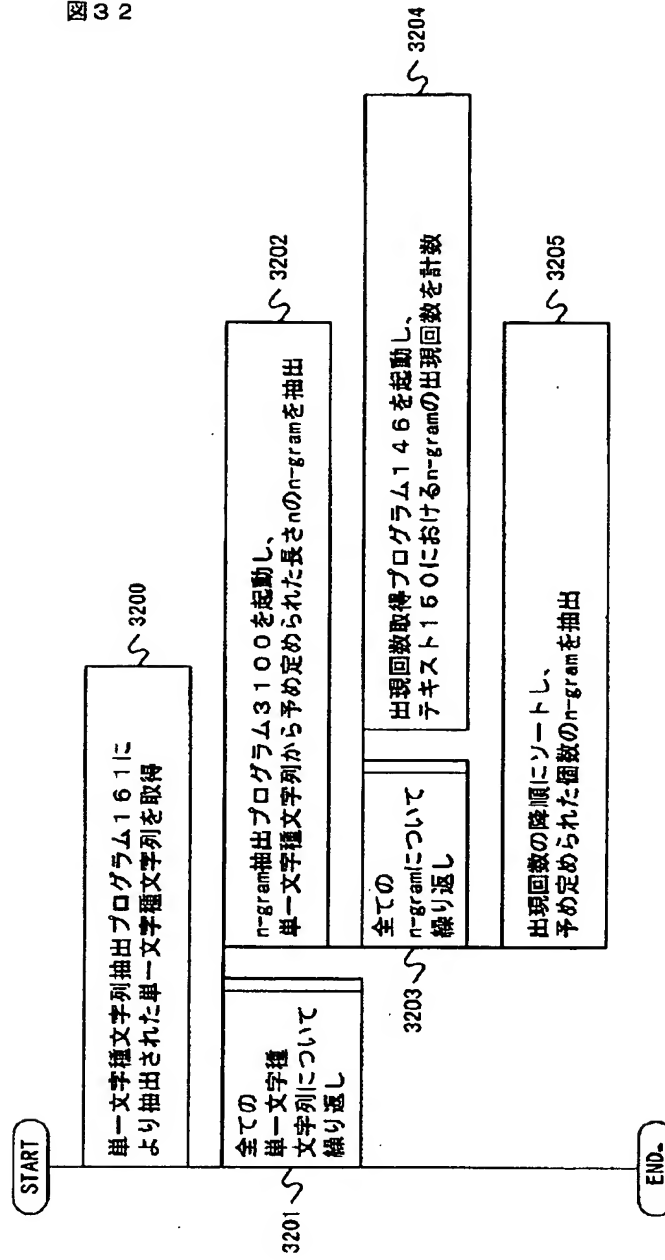
図27





【図32】

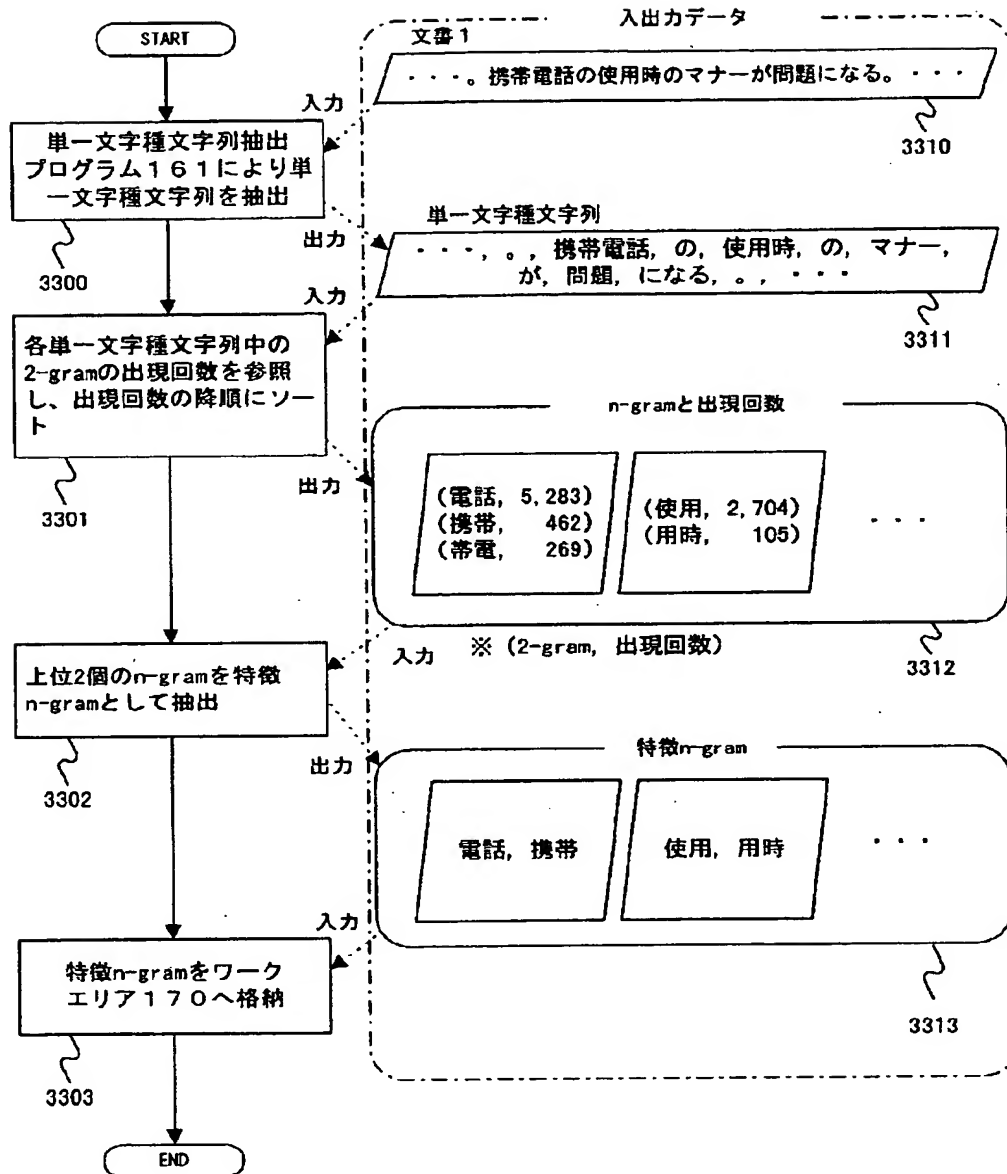
図32





【図33】

図33



フロントページの続き

(72)発明者 菅谷 奈津子  
 神奈川県横浜市都筑区加賀原二丁目2番  
 株式会社日立製作所システム開発本部内

(72)発明者 川下 靖司  
 神奈川県横浜市戸塚区戸塚町5030番地 株  
 会社日立製作所ソフトウェア開発本部内